Think

October/November 1974

THE BIG BOOM IN BICYCLES

Page 4

A SPECIAL REPORT ON PROGRAMMING

Page 18

Letter from the Chairman

We can all fight inflation

Our business may seem to be pointing in two different directions these days. On the one hand, thanks to your efforts, IBM is doing well. After record results for 1973, we have just chalked up the best nine months of gross income and earnings in our history. On the other hand, inflation and increasing signs of business recession are causes of deep concern to all of us.

In these contradictory circumstances, prudent business judgment requires that we take whatever steps are necessary to help weather the seas ahead.

One way is to make special efforts to control costs, to do our part to help slow inflation. The goal is to cut out needless expenditures and to make sure we get—as well as give—full value for every dollar.

On another front, in view of the many uncertainties in the economy, we have temporarily halted additional employment. This is a cautionary measure which we will review periodically as the economic situation becomes clearer. It should be noted that we have already hired some 11,000 additional people this year in the United States, and a similar number in World Trade, and these should be adequate for our needs at this time.

As individuals, the most important thing each of us can do to help fight inflation is to increase productivity. Some IBMers are already doing a great deal on this front. One recently made an innovative suggestion that reduced the testing time of a product by 54 hours per unit. Another suggestion to substitute computer facilities for the hand-posting of records saved one of our branches nearly 3,000 hours of administrative time annually.

We need hundreds more imaginative actions like these to hold down costs and increase output and get us through the challenging times ahead. I am confident we can come up with them if we will all put our minds to it.

Frank Cary

Features

Chairman's letter

Notes 3

Work style 16

People 26

Think
Volume 40 Number 7
October/November 1974

4 **The big boom in bicycles**
Schwinn and Peugeot wheel out the computer to help make it go.

12 **Education, Italian style**
Novedrate. Its graduates include more than 5,000 IBMers and customer executives.

18 **Programming**
A special report on where it's at and where it's going.

- **Primer on programming**
- **State of the art**
- **Programming languages**
- **Visit to a programming center**
- **Outlook**

36 **Exercise—who needs it?**
A scientific look at the physical fitness phenomenon.

40 **Digging into history**
Archaeologists use the computer to analyze artifacts.

44 **Perseverance**
The handicapped need it to succeed. A new equal opportunity law will help.

46 **Benefits. Your hedge against inflation**
Last year, IBM's benefits bill averaged $5,043 per U.S. employee.

How IBMers keep fit 38

Speak Up anniversary 43

©Copyright 1974 by International Business Machines Corporation. All rights reserved.

Think

The contents of this issue, in brief

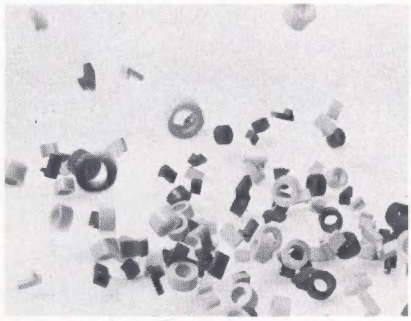
THE COMPANY

Benefits: A hedge against inflation 46

To keep up with rising costs and to provide wider coverage, IBM last year spent more on benefits per U.S. employee than it spent for research and development or paid out in stockholder dividends. The result: less out-of-pocket expense for the typical employee in a period of rapidly rising inflation.

The cost of medical benefits, to cite a major expense, has jumped 132 percent since 1969. Yet the company's programs have been improved, despite rising costs. Not only do active employees gain, but average retirement income has risen 80 percent since 1969.

TECHNOLOGY

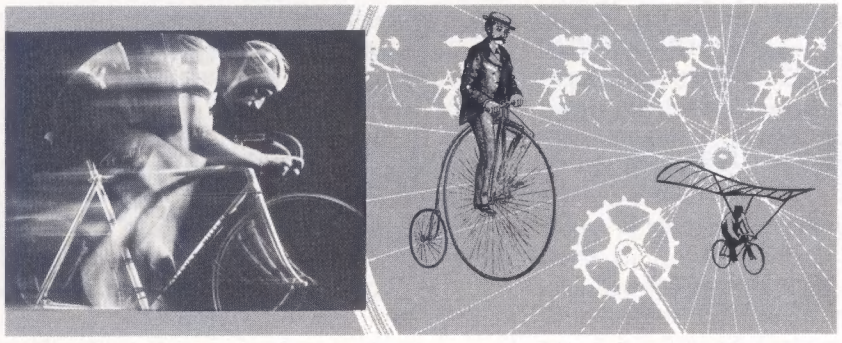


Programming: A Special Report 18

Just 25 years ago, computer programming was practiced by a handful of scientists and technicians. Today, it is the livelihood of hundreds of thousands of people around the world, working in government, education, and just about every kind of business. It is also one of the biggest cost items in the $24-billion U.S. data processing budget.

This Special Report focuses on programming, what it is, and where it's going. A professor of computer systems at the State University of New York takes a novel look—in layman's language—at the practice of programming. An expert on programming languages describes some of the problems involved in improving communication techniques between the programmer and the computer. Top IBM executives and a customer reflect

THE MARKETPLACE



Big boom in bicycles 4

The current world bicycle boom might have astonished even Baron Karl von Drais, the German nobleman generally credited with inventing the two-wheeled vehicle in 1816. Not since the turn of the century when citizens took to the wheel in uncounted numbers has there been anything like it. The current bicycle frenzy has even taken bicycle makers by surprise.

Two bicycle manufacturers prominent, then as now, are the family firms of Schwinn in the U.S. and Peugeot in France. Both count heavily on handcraftsmen to turn out their racing machines. Both are making increasing use of IBM computers to stay on top of a business in which styles can change overnight and critical parts are often in short supply. "There are fantastic changes to keep track of today," says Frank V. Schwinn, president of the company that bears his name. "This means it's absolutely imperative to have reports with figures you can touch right now—not tomorrow."

ISSUES

Exercise—who needs it? 36

What is the real effect of exercise? Does it improve health? Prevent heart attacks? Increase longevity? For years, answers to these questions have come largely from athletic coaches and exercise buffs themselves. In recent years, however, exercise has been given careful medical scrutiny. One of the most significant findings, reports science writer Lee Edson: "cardiovascular fitness"—the efficiency of oxygen pumping in the body—improves as a result of regular, vigorous physical activity.

In an accompanying article, IBM

For nonprogrammer readers who may wonder what programming is all about, *Think* presents an easy-to-read primer on the subject. In it, the author likens the program to a recipe for a computer-like automatic cake-baking machine. The analogy begins with eggs—six large eggs.

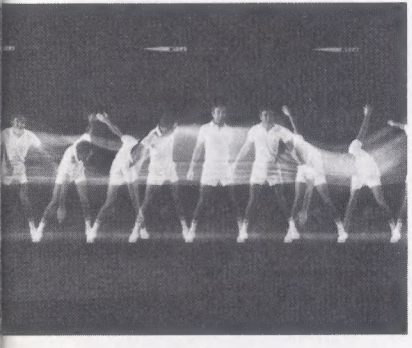
Page 18

on a future that promises direct communication between the layman and the computer terminal.

Additional articles report on new techniques that are improving the programming process. The goal: to bring software costs in line with cost reductions already achieved in hardware through advanced technology.

A visit to the New England Programming Center and profiles of five practitioners of the art show the wide variety of people engaged in programming around the globe.

Medical Director Dr. John Duffy encourages employees to follow a regimen of regular exercise on their own time. Voluntary programs have already begun at management schools. But if exercise is to have any lasting value, explains Dr. Duffy, people have to pursue their own after-hours exercise program.

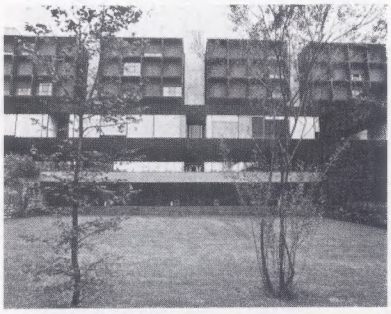


THE WORLD

Education, Italian style **12**

It's a classic contemporary building that serves a pragmatic purpose. Since opening last January, the new education center in Novedrate, 20 miles north of Milan, has held classes in data processing for more than 5,000 IBMers and 400 customer executives. Novedrate is IBM's largest residential education facility in Europe. It offers more than 100 courses.

Accompanying this article: an assessment of the business outlook by the president of IBM Italy, and a brief report on overall IBM education efforts abroad.



HUMANITIES

Digging into history **40**

Dental picks and brushes are still essential tools for archaeologists laboring to uncover the past. But, increasingly, the computer is being used to describe, classify, and analyze the artifacts that are found. Dr. Donald Fisher, director of Oklahoma State University School of Mathematical Sciences and professor of computer sciences, tells how he helped build a comprehensive computer data base while unearthing the ancient city of Caesarea Maritima on the coast of Israel. Drawing on the computer coding system, Dr. Fisher is able to describe 186,000 distinct rim shapes for various kinds of pottery that have been discovered.

Notes

President Ford's recent call for a "new mobilization against inflation" culminated weeks of preparation that included meetings with businessmen and economic experts from around the nation.

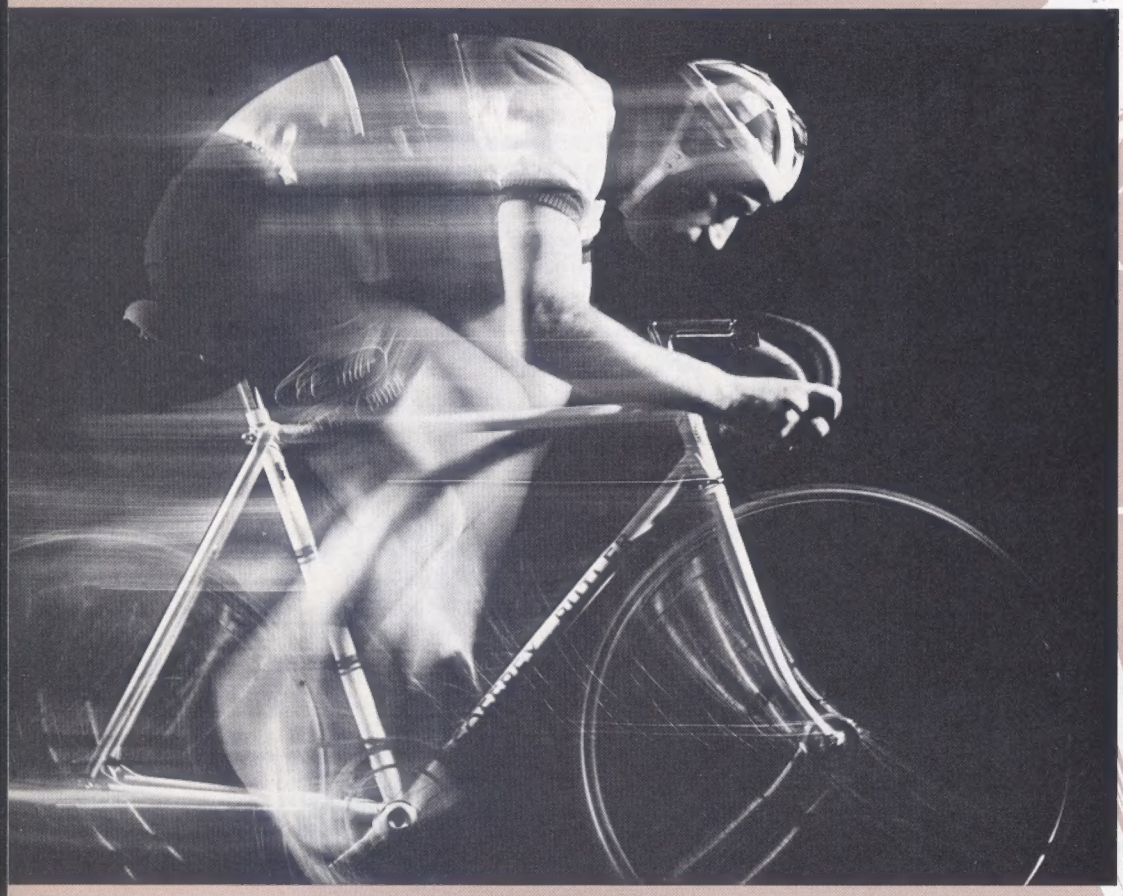
Among the businessmen and economists invited to the "mini-summits" convened in September by President Ford were IBM Chairman Frank T. Cary and Dr. David L. Grove, IBM vice president and chief economist. They joined in the search for new ideas to deal with U.S. economic problems, especially inflation.

Cary called for a moderate easing of the government's monetary policy and a cut in Federal expenditures. "The present monetary policy," he said, "is depressing the stock and bond markets, impacting the housing industry and thrift institutions, and making it very difficult to finance additional capacity in the primary industries."

He recommended tax relief for lower income people and tax incentives to encourage more investment in the primary materials industries and utilities. "In order to balance this tax relief," said Cary, "it may be necessary to increase taxes on higher incomes and corporations. If it is necessary, we should step up to it."

Grove forecast very little growth throughout next year and "a sharp deterioration in corporate profits as we enter 1975." He said he expected the rate of inflation "will taper but only very slowly," and predicted that by the fourth quarter of next year the Consumer Price Index will be rising at a rate of about 7.5 percent. The real danger, said Grove, is a "wage explosion that feeds on itself," forcing inflation figures even higher.

"There isn't any easy solution," he declared. He said wage and price controls would not work under the present circumstances. And he emphasized that all segments of society must contribute to a solution, with no one group increasing its share of national income "to any considerable extent over the next year or so."



THE BIG BOOM IN

BICYCLES

by Geoffrey D. Austrian

The cyclists pour through the park in a seemingly endless stream—a blur of brightly colored jerseys and silvery wheels against the fall foliage. It's Sunday in New York's Central Park and the roads are banned to auto traffic. But it could be Chicago's Lincoln Park or Golden Gate Park in San Francisco.

By all accounts, there's been nothing like it since the great bicycle craze of the 1890s. For the first time since the motorcar swept away the happy halcyon days of cycling, more bikes are being sold than cars—in the U.S. alone, an estimated 16 million during 1974, compared to fewer than 10 million automobiles.

This resurgence of the bicycle has taken even bicycle makers by surprise. There are briefcase-bearing businessmen, bikecologists seeking cleaner air, entire families out for the sheer fun of it. In Davis, Calif., 40 percent of the citizens have taken to cycles. Suddenly, it's the thing to do. One-upmanship is no longer a larger gas-guzzling car—or, for that matter, a fast, sleek compact. When your neighbor circles the block on a 10-speeder, take after him on a 15-speeder, *The Wall Street Journal* advises.

More than $24-million in highway trust monies, formerly a sacrosanct cache, were released last July to 24 states for bike paths as well as hiking trails. But states had already gone ahead on their own, putting in more than 25,000 miles of bikeways in the past dozen years.

The bike boom is not exclusively a U.S. phenomenon. In France, "Le Cyclisme" has been welcomed back with a passion. Bicycle sales have doubled since 1969. This year, the government promoted a national bicycle day, and voted additional francs for bikeways. Rent-a-bike businesses have sprung up at railway stations from Normandy to the Côte d'Azure.

The modern-day bike fad, almost certainly, would have astonished Baron Karl von Drais, the German nobleman generally credited with

having produced the first of these two-wheeled vehicles. His "draisene" or "hobby horse" was a crude two-wheeled contraption, propelled forward by pushing your feet backward along the ground. But it caught on quickly when it appeared in 1816. Soon, well-to-do citizens were racing hobby horses up and down the boulevards of European cities. By 1839, a practical Scotsman named MacMillan added pedal cranks and soon a Frenchman, Pierre Michaux, began to mass produce the "bone shaker"—so called because it ran on wooden wheels.

However, bicycle technology appeared to take a giant leap backward with the British "penny-farthing." It gained its name because the large British penny and tiny farthing coins reflected the relative sizes of the front and back wheels. The larger the front wheel, it was reasoned, the faster the bike could go—if the rider could summon the energy to turn the pedals. Perched high over the front "penny" wheel, the rider was in a precarious position. A bump in the road could mean "taking a header," or forward spill.

Though the "big wheels" remained popular for some time, it took the chain-driven "safety" bicycle, with the rider seated in relative safety between the two wheels, to get cycling back on the track. Early riders, however, faced hostility when they hit the road. Farmers stuck pitchforks between the spokes and blocked the rights of way with wagons. Horsemen charged at cyclists

They go first class on the product line—with the help of computers

Schwinn

for the fun of it. Later, motorcar enthusiasts made sport of brushing the cyclists aside. To make matters worse, innkeepers refused to serve women cyclists wearing bloomers.

Bicycle riders banded together for self protection and to campaign for better roads. In the U.S., the League of American Wheelmen, formed in 1880, enlisted 100,000 members within a score of years, including such prominent members as Commodore Vanderbilt and the Wright brothers.

Bicycles not only filled the nation's byways—two million bikes were sold in 1897—they also caught the public fancy. Physicians saw them as a boon to good health. When crack shot Annie Oakley splintered glass targets in Buffalo Bill's Wild West Show, she did so not from a horse—but from a bicycle. And some military minds saw the benign bicycle as the ultimate weapon of a mobile army.

The bike, "an ever-saddled horse that eats nothing and requires no care," gave citizens a new freedom. But the best rationale for the bicycle mania—then and now—was given recently by Schwinn Bicycle Company's Cycling Activities Manager Keith Kingbay, a 61-year-old sportsman who turns out for occasional senior races and bikes some 8,700 miles a year: "The world," he says, "looks different to you from the seat of a bicycle."

Different?

"Until you've tried it, you don't know what you're missing out of life."

The year 1895 was a troubled time. The U.S. was struggling to work its way out of a severe financial panic. The business outlook was bleak. War with Spain loomed on the horizon. But despite the worries—or perhaps to escape from them—nearly everyone was buying a bicycle and singing a new song about Daisy and her "bicycle built for two."

Three hundred bicycle factories and countless small assembly shops struggled to meet the demand. "It was a little like it is today," says Frank V. Schwinn, president of the Schwinn Bicycle Company. "Everyone was trying to get into the bicycle business."

It was also the year Schwinn's grandfather, Ignaz, a German-born bicycle mechanic, set up shop in Chicago to turn out his own design of a "safety" bicycle, so called because it was a lot more stable than the high-wheelers. The energetic craftsman (he went to work every day until he was 85) started a tradition—colored by an Old-World love for perfection—that has endured to the present day.

Schwinn Bicycle Company is neither the oldest nor the largest in the U.S. It was a Civil War cavalry officer named Albert A. Pope who first put the U.S. on wheels after being impressed by a display of British high-wheelers at the Philadelphia Centennial Exposition of 1876. Colonel Pope's *Columbia* bicycle, since acquired by another firm, is still produced today. As for volume, three U.S. manufacturers—Murray Ohio, Huffman, and AMF—ranked ahead of Schwinn in 1973, according to *Fortune* magazine. But, in many ways, Schwinn, which *Fortune* calls "the aristocrat of the industry," is considered by many to be something special in the U.S. market.

Super balloon tire bikes, such as this 1950s Black Phantom *(foreground), which housed horn and batteries in a streamlined tank, were sales leaders for Schwinn for more than 30 years. Behind it, President Frank V. Schwinn leans on a* Varsity *model, the company's most popular 10-speeder today. Founder Ignaz Schwinn, who introduced his own design of safety bicycle in 1895, looks on approvingly from portrait on wall.*

Schwinn, its president points out, markets bicycles exclusively under its own name, a factor that he says places it squarely behind its product through service dealerships and a lifelong guarantee of all parts. While most U.S. bike makers started as family companies, Schwinn has continued as one, with the company's stock held in a family trust. The resulting feeling of tradition—but not of stuffiness—that pervades the family firm engenders a strong sense of proprietorship.

"We build a bicycle the way it should be made," says Frank Schwinn. "We're not basically looking at what the competition is doing. We're looking at how we can make a bicycle last longer and perform better."

It's the same long view expressed in an advertising poster for the *World* bicycle, an early Schwinn product on which the company's racing teams set many records. "For men may come and men may go," declaims a Tennyson-inspired versifier, "but we *roll* on forever."

On October 22, Schwinn rolled past the 80-year mark. But not without some bumps along the way. The first bicycle boom collapsed around 1900, taking many firms with it. Although Ignaz Schwinn diversified into motorcycles, it was his son, the late Frank W. Schwinn (father of the current president of the company), who saved the business by turning the bicycle into a children's toy.

"My father introduced the balloon tire in 1933," recalls Schwinn. "Before that, all tires had a single tube. When you ran over a nail or a piece of glass, all you could do was squeeze some gooey substance called 'neverleak' into it. And the tire leaked, or didn't leak, depending on the size of the hole. Keeping a bike was an expensive proposition."

Although not an engineer by training, Schwinn attributes each new plateau in the bicycle business to technical innovation. "The development of the 10-speed bicycle," he comments, "has opened up bike riding to many more people. The better gear systems make it possible for all kinds of people—old and young—to get equal enjoyment."

While developing a new market for children's bicycles, the Schwinn Company never gave up on fine lightweight bicycles for adults. It introduced the first 10-speeder to the U.S. in 1938. That model, the completely handmade *Paramount* remains at the top of its line, selling for about $575. Every four years, Schwinn furnishes *Paramounts* for the U.S. Olympic team. But the company's first production model 10-speeder, brought out in 1955, failed. "Our own dealers thought it was impractical to sell such complex equipment," comments Ray Burch, Schwinn's vice president of marketing. "But as we ran training schools and improved dealer service, we gradually won acceptance."

The 10-speed revolution anticipated by Schwinn quite literally pushed the bicycle business down a new path. In 1968, the company sold more than a million bicycles for the first time—85 percent of them the distinctively styled children's bicycles with banana seats, high-rise handlebars, and smaller front wheels.

Three years later, following a temporary dip in sales, Schwinn again shot across the one-million mark. But this time 85 percent of its bicycles were built for adults. And the typical bike was a trim 10-speeder. From a fairly stable industry, where everyone knows what's happening, the bicycle business found itself in a dynamic new market.

"There are fantastic changes to keep track of today," says President Schwinn, "and not nearly as much time to think a problem through. This means it's absolutely imperative to have reports with figures you can touch

right now—not tomorrow."

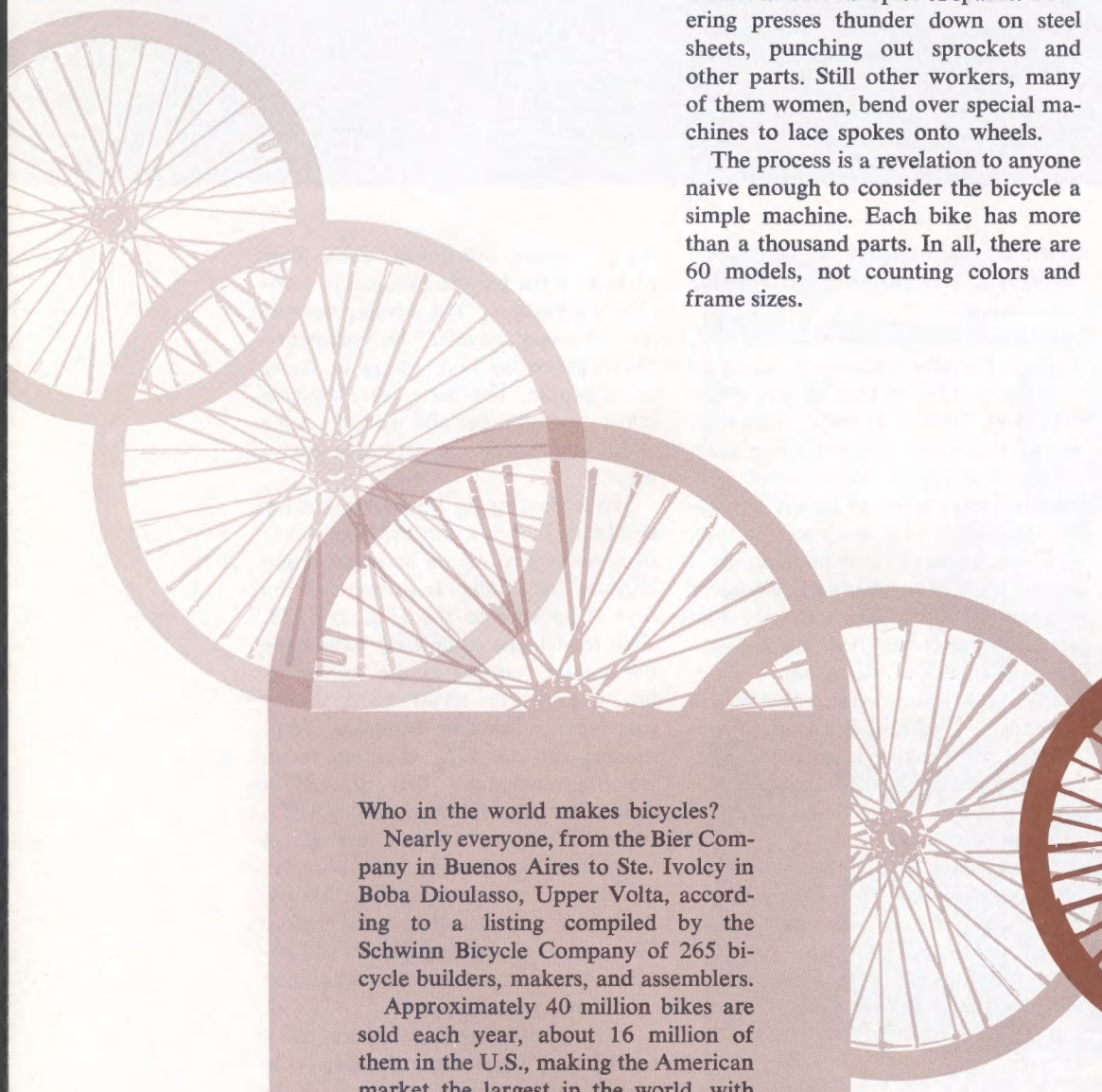
The dynamism is seen in Schwinn's Chicago plants, which turn out some 6,700 bicycles a day. A bicycle factory is a lively happening. Miles of tubing spurt from forming machines. Workers bend it into handlebars and weld it into frames amidst canopies of sparks. Towering presses thunder down on steel sheets, punching out sprockets and other parts. Still other workers, many of them women, bend over special machines to lace spokes onto wheels.

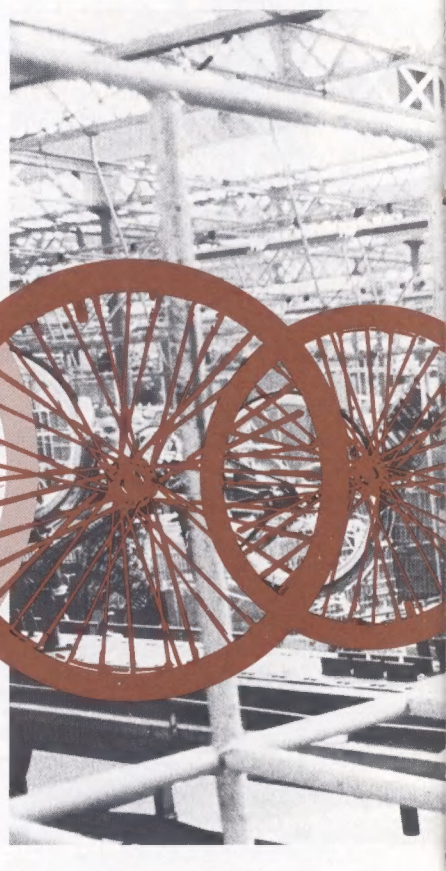
The process is a revelation to anyone naive enough to consider the bicycle a simple machine. Each bike has more than a thousand parts. In all, there are 60 models, not counting colors and frame sizes.

Schwinn started to manage this multiplicity by computer in 1968 when IBM equipment applied to accounting was turned to inventory control.

"We had been using manual 'hot lists' for many years to keep track of parts in short supply," says Al Selkurt, senior vice president and treasurer. "But they didn't help us spot surpluses in our 4,500 parts when the model mix changed. I knew how much was over in terms of total investment, but it took a great deal of manual analytical work and physical inventory work to learn what to get rid of. It's a lot easier today."

Schwinn's toughest problem in a surging market had been determining the long lead time required for foreign





PEUG

Who in the world makes bicycles?

Nearly everyone, from the Bier Company in Buenos Aires to Ste. Ivolcy in Boba Dioulasso, Upper Volta, according to a listing compiled by the Schwinn Bicycle Company of 265 bicycle builders, makers, and assemblers.

Approximately 40 million bikes are sold each year, about 16 million of them in the U.S., making the American market the largest in the world, with one-quarter of the bikes sold in the U.S. coming from abroad.

While the U.S. buys the most bicycles, it does not have the largest number of bicycle makers. France tops the list with 30, followed by Belgium with 27, and Japan with 18. Others are the United Kingdom and West Germany, each with 17, and Italy with 15. The U.S. has nine bike manufacturers.

The U.S.S.R., it is estimated, produces some four million bikes a year—all made by one government-owned firm—China makes half that many under the names of *Flying Pigeon* and *Phoenix*.

In contrast to most of the world, where the bicycle is regarded as an essential mode of transportation, approximately 95 percent of the bikes in the U.S. are sold for recreational purposes.

by Rupert Deighton

components. Brakes come from Switzerland; derailleurs (the gear-changing systems) from France; wheels from Belgium; pedals from West Germany; and saddles from Great Britain. Though many parts are made abroad, they are made to Schwinn specifications and many are turned out exclusively under Schwinn patents.

Jim Banks, an advisory systems engineer in the Data Processing Division's Chicago West branch office, worked with the Schwinn data processing staff to solve the problem. "To make a bike of a certain model, you need a frame, you need handlebars, grips, fenders, and other components," he explains. "You specify to the computer how many and what kinds of parts you need at each stage of assembly. In other words, if you want a completed bike in May, you need frames in April and the parts that make up the frame in March. So you get a March, April, May sequence of events.

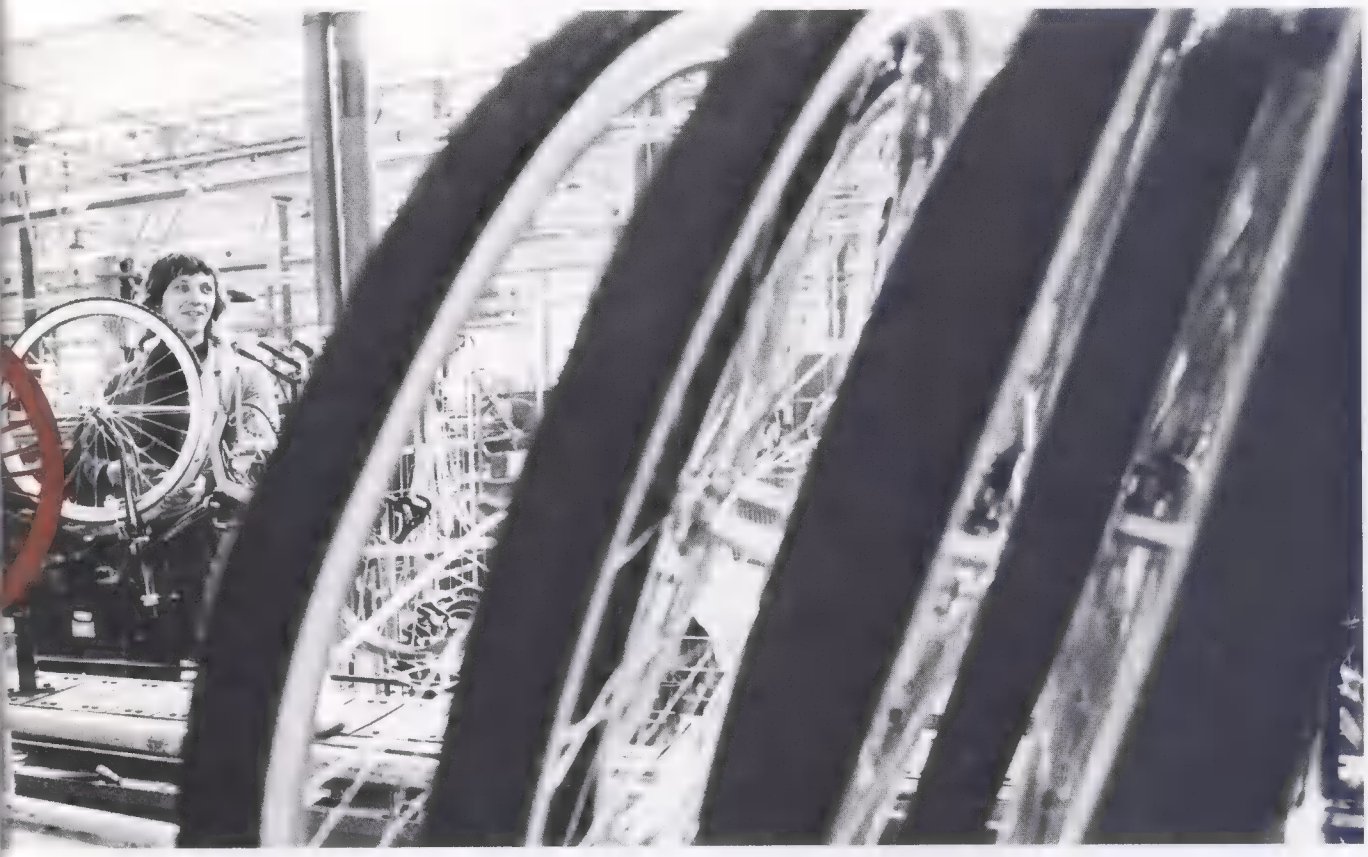
"The application we installed is called time-phased bill of material planning. The computer lets Schwinn know how far ahead to order parts. And they either issue purchase orders or delay them to make up their production plan. They can replan in a few hours."

Through data processing, Schwinn has been able to increase its lead time to vendors, says Selkurt, and determine much earlier when potential shortages are going to arise.

Schwinn's data processing manager, Joe Heilingoetter, is busy tripling his staff to tackle other applications. One incentive: Newly enacted Federal safety legislation requires manufacturers to keep track of parts going into particular products. Computers can simplify a recall, should the need arise.

To DPD Marketing Representative Soo Young Suh, who began his career three years ago in IBM Korea where he sold the first System/370 Model 135, Schwinn has become a customer "with lots of potential."

Marketing Manager Jack Winter agrees: "With Schwinn's long tradition for quality, you can be sure that when they decide to do something, they do it in a first-class way." ■



Leaning lazily, a stack of wheels waits to be assembled onto Peugeot New Style *bicycles moving along the line. The folding portable can easily fit into a car trunk or a closet.*

PEUGEOT

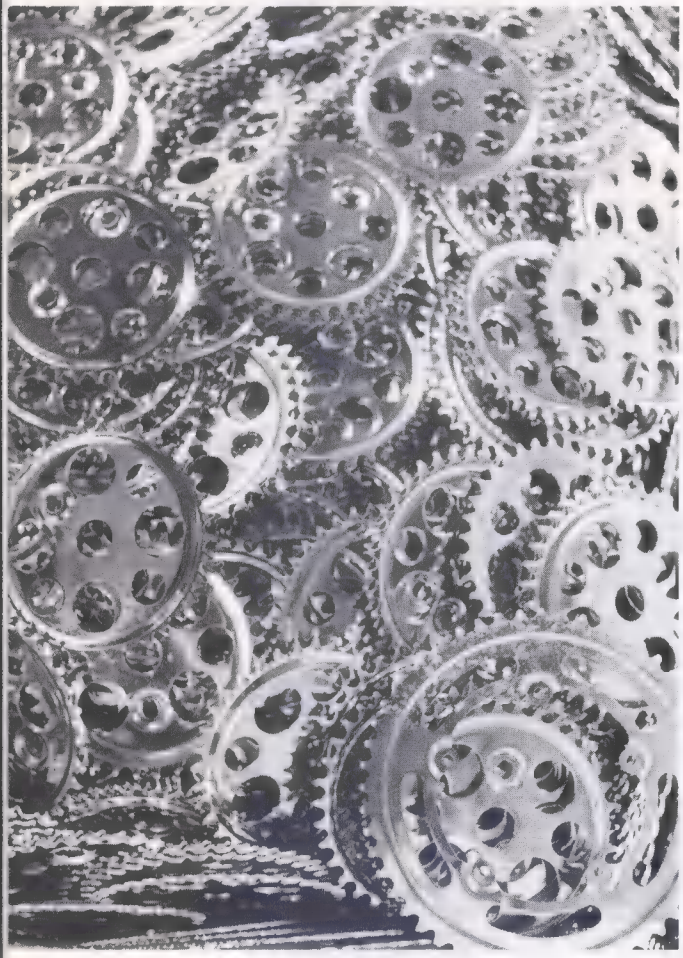
It's the number one bike maker in France, a System/370 user

In France they call it *La Renaissance de la Petite Reine,* the rebirth of the little queen, or the bicycle, so long dethroned by the majesty of the automobile. Today, nearly 11 million cycle-owners of all ages are pedaling the highways for health and pleasure or wheeling their way to work in such auto-choked cities as Paris and Lyons.

Behind this resurgence—not only in France but across the world—is an old and respected family firm, perhaps better known for its automobiles, the Groupe Peugeot S.A. While Cycles Peugeot accounts for less than 10 percent of the group revenue—around $155-million last year—the figure has been climbing 20 to 25 percent a year. Sales in France and overseas leapt from 184,000 in 1964 to over 785,000 last year, establishing Peugeot as France's number one manufacturer and exporter of bicycles.

"We could have sold another 140,000 machines this year, if we had been able to produce them," says Roger Saulpic, commercial director for France. "But we prefer to meet realistic targets and satisfy our customers completely."

While the head office of Cycles Peugeot is in a new building in a fashion-



A typical bicycle has more than a thousand parts such as the sprockets in this basket. With a growing number of models and rapidly changing styles, manufacturers are increasingly relying on IBM computers to manage their inventory and production.

able quarter of Paris, the heart of the business is in its traditional home some 400 kilometers to the southeast in the little town of Valentigney on the heavily wooded banks of the river Doubs. Some 6,000 of the 9,000 company employees work in the factory there, with others at a new factory at Romilly, less than 100 kilometers from Paris, and at Saint-Louis in the Haut-Rhin Department where the motors for cyclomoteurs, small motorcycles, are manufactured.

It was to Valentigney that the founder of the French bicycle industry, young Armand Peugeot, returned in 1871, after studying in England, to join the family metallurgy business. In England he had been bitten by the "bike bug" and he now launched a 12-year campaign to convince his father and brother that they should extend their business into the new era of leisure and transport. In 1885 he got his way, and Peugeot produced the first of a long line of durable, high-quality cycles.

At this time, Continental manufacturers such as Peugeot tended to concentrate on leisure cycles while British manufacturers developed the long-standing tradition for heavy, robust machines that were an economical form of transport for the growing population of industrial workers. Peugeot, meanwhile, chalked up its first success in cycle racing in the 1890s with Marcel Bourillon winning the world cycling championship. This established the name of Peugeot in the public eye, and, by 1900, over 20,000 machines a year were being produced at the Valentigney factory.

In those early days, Peugeot recruited employees from the local region. Less than 20 miles from the Swiss border, the company found part-time farmers, who during the winter months were accustomed to making the famous Swiss "cuckoo" clocks. The staff numbered around six or seven hundred. By 1968 this had only risen to 3,200. The explosion in demand has swelled this number to over 9,000.

Today Peugeot produces four families of bicycles—the *Classic, New Style, Weekend,* and *Sporting*—for which there are 274 listed styles and variations.

Each day, under the watchful eye of Manager Yves Saugier, the 1,000-strong cycle shop work force, over 30 percent of whom are women, turn out some 2,600 machines. Building a cycle takes, on average, only three and a half hours, from the moment when the rear forks are welded together to the time when the assembled machine is dropped into a heat-sealed plastic bag, ready for delivery. All bear the hallmarks of Peugeot's sporting tradition with lightweight frames, up to 10 or 12 gears and, on all but the *New Style* (a folding portable), pencil-thin wheels and racing-style dropped handlebars.

Helping to guide plant operations by ensuring that sufficient materials are on hand and by tracking the use of components throughout production is an IBM System/370 Model 135 installed earlier this year. Since Peugeot started using IBM equipment for routine accounting applications in 1969, annual production of bicycles and cyclomoteurs has rocketed 250 percent, with more than 1.2 million units shipped each year.

"We could not have handled such an increase in demand," says Gilbert Schodt, head of Peugeot's data processing department, "if we had continued using manual production control methods.

"We have over 120,000 different components for production here or that are manufactured for other Peugeot Group plants or for shipment to other companies. Each day we ship some 25,000 items. So control has to be tight. The System/370 is a crucial part of our operations. Without it, we would probably come to a halt in a matter of days."

Daniel Perrot, one of the IBM marketing representatives for the combined Peugeot Group account, further explains: "Every day, the computer is used to determine the exact number of parts to be manufactured. Not only

that, but it also keeps track of work in progress."

Tied to each bicycle frame in a cellophane bag is a slim sheaf of punched cards. At each work station, additional information is entered into the cards and relayed back to a file in the central computer. The central computer is also linked with Peugeot's new cycle factory at Romilly, where 1,350 models a day are assembled from parts delivered from Valentigney.

While much of production is automated at Valentigney, hand finishing remains important, according to Production Manager Saugier. And only five entirely handmade *Prestige* cycles are produced each day for professional and amateur racing cyclists—and for connoisseurs from all over the world.

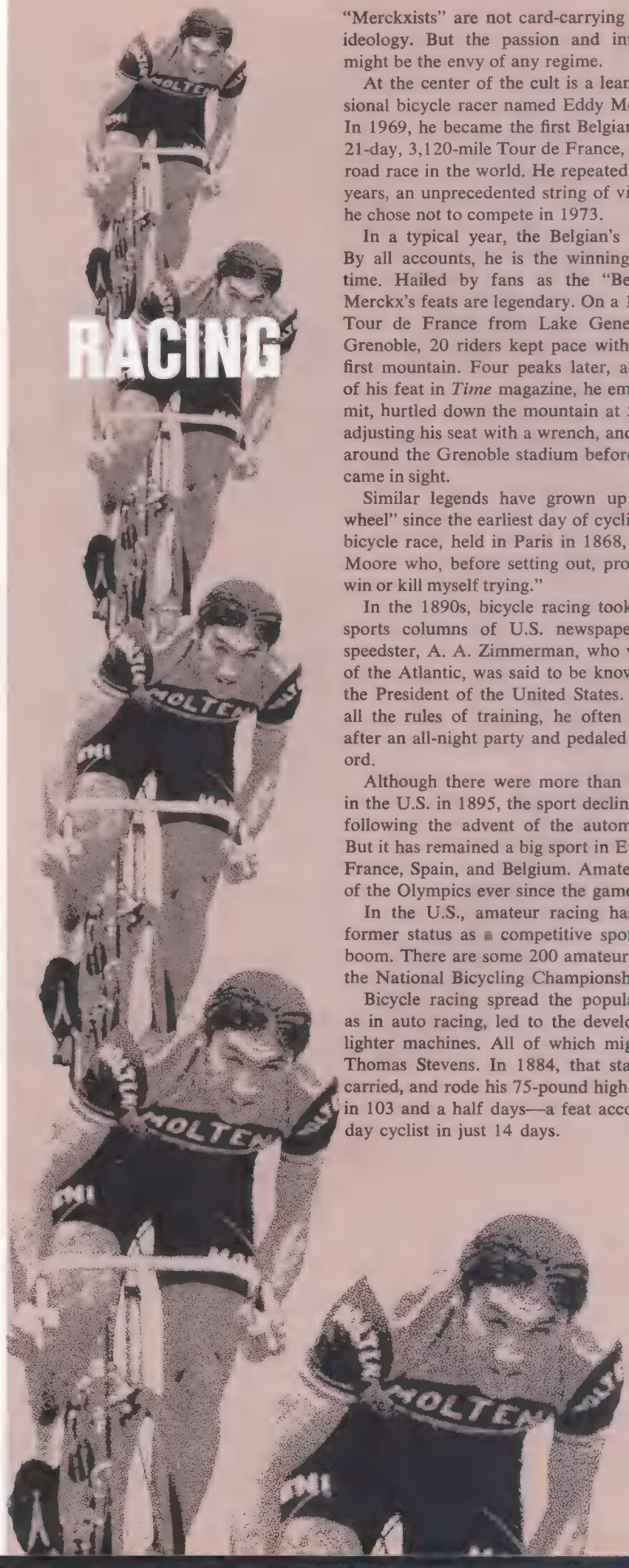
For most models, one of every ten parts is tested; while for sporting models, each part is tested. More than 250 testers are on hand to ensure nothing gets left to chance.

Such attention to detail is much appreciated in export markets where availability of spare parts can be a problem. Peugeot's export director, Henri de Laissardiere, counts it as one of the major reasons for Peugeot's success in America.

Exports play an increasing role in the Peugeot story. Almost half of total production goes abroad or is manufactured under license overseas. The U.S. accounts for 140,000 machines annually, over 30 percent of exports. In 1960, the figure was just over 1,000.

"The reasons are not hard to find," contends M. de Laissardiere. "America has discovered the French bicycle. It is lighter and easier to ride. In short, it is more fun. And of course, the Americans pay attention to what their doctors tell them. The sedentary urban life is extracting a terrible toll on human health." ■

A British IBMer, *Rupert Deighton is a senior writer at the Communications Support Center of* IBM *France in Paris. He is also a bicycle buff whose grandfather happened to be head of Raleigh Industries, the British bicycle maker.*



"Merckxists" are not card-carrying followers of a political ideology. But the passion and intensity of their beliefs might be the envy of any regime.

At the center of the cult is a lean, Latin-looking, professional bicycle racer named Eddy Merckx (pictured at left). In 1969, he became the first Belgian in 30 years to win the 21-day, 3,120-mile Tour de France, the most famous bicycle road race in the world. He repeated the feat four successive years, an unprecedented string of victories that ended when he chose not to compete in 1973.

In a typical year, the Belgian's earnings top $350,000. By all accounts, he is the winningest bicycle racer of all time. Hailed by fans as the "Beethoven of the bike," Merckx's feats are legendary. On a 120-mile leg of the 1970 Tour de France from Lake Geneva across the Alps to Grenoble, 20 riders kept pace with him to the foot of the first mountain. Four peaks later, according to an account of his feat in *Time* magazine, he emerged alone at the summit, hurtled down the mountain at 50 miles per hour while adjusting his seat with a wrench, and made a ceremonial lap around the Grenoble stadium before his nearest competitor came in sight.

Similar legends have grown up around "kings of the wheel" since the earliest day of cycling. The first recognized bicycle race, held in Paris in 1868, was won by Dr. James Moore who, before setting out, proclaimed: "I shall either win or kill myself trying."

In the 1890s, bicycle racing took up more space in the sports columns of U.S. newspapers than baseball. One speedster, A. A. Zimmerman, who won titles on both sides of the Atlantic, was said to be known to more people than the President of the United States. Notorious for violating all the rules of training, he often climbed on his bicycle after an all-night party and pedaled his way to another record.

Although there were more than 600 professional racers in the U.S. in 1895, the sport declined rapidly in popularity following the advent of the automobile and auto racing. But it has remained a big sport in Europe—notably in Italy, France, Spain, and Belgium. Amateur racing has been part of the Olympics ever since the games were revived in 1896.

In the U.S., amateur racing has regained some of its former status as a competitive sport with the new bicycle boom. There are some 200 amateur races a year, topped by the National Bicycling Championship.

Bicycle racing spread the popularity of the bike. And, as in auto racing, led to the development of stronger and lighter machines. All of which might have impressed one Thomas Stevens. In 1884, that stalwart dragged, pushed, carried, and rode his 75-pound high-wheeler across the U.S. in 103 and a half days—a feat accomplished by a modern-day cyclist in just 14 days. ■

Education Italian Style

A country estate near Milan has been turned into the largest residential education center in IBM Europe. So far, over 5,000 IBMers and customer executives have studied there.

by Angelo M. Donofrio

On the road to Novedrate, 20 miles north of Milan, you can see the new IBM education center through the treetops. The four-story, classic contemporary structure of glass and concrete—with one floor of classrooms and three of dormitories for resident students—stands among the rolling hills of a 45-acre estate that achieved some prominence during the 19th Century. It was owned by Count Carlo Taverna, head of a noble family of Lombardy, who helped the Italian independence movement in 1848.

Nearby is the tiny village of Novedrate (pop. 1,880), famous for its furniture-makers, weavers, and wood carvers, and surrounded by green hills and solemn cypress trees. Ten miles to the north is scenic Lake Como and the foothills of the Italian Alps.

Count Taverna's stone villa with its sloping red-tile roof still stands amid the green English-style park, now the grounds of the education center on Via Isimbardi. It is still used occasionally for IBM management meetings. But the heart of the IBM operation in Novedrate is the imposing education center designed by architect Bruno Morassutti. Since it opened last January, more than 5,000 IBMers—primarily from IBM Italy—and 400 customer executives have attended classes there.

Novedrate is IBM's largest residential education center in Europe. It can house 240 IBMers and 24 customer executives at the same time, plus an additional 160 IBMers on a nonresidential basis (mostly commuters from Milan).

Giancarlo Bonghi, the country's DP education manager, says the new center "creates a college atmosphere which encourages learning and allows students to arrange their classwork, study time, meals, and leisure with complete freedom."

One recent example: A group of high-level diplomats in the customer executive program stayed up until one in the morning, solving problems on

Novedrate can accommodate 240 IBMers and 24 customer executives on a residential basis. The goal of the architecture is to blend the pragmatic needs of a modern education center with the hilly, richly wooded countryside.

their computer consoles, "despite the fact that practically every other Italian was watching a championship soccer match on television," says Mario Dutto, customer executive department manager.

"There are still top executives from many companies who do not understand the computer," he adds. "We offer them courses in basic data processing concepts, how computers affect a company's organization, applications such as business planning, plus a number of courses on how computers can help top management in the decision-making process."

Altogether, Novedrate offers more than 100 courses for salesmen, systems engineers, and customer engineers, as well as customers. Classrooms are linked by display terminals to the center's computer room and to an audio-visual studio equipped to transmit closed-circuit television.

At the request of the Italian Government, special IBM courses have been set up for national and municipal officials. In addition, 150 teachers, professors, and university deans have taken courses at Novedrate, and graduate students have received on-the-job computer training there. A television course on computing, prepared with help from Novedrate, was shown nationally dur-

A capsule view of IBM Italy

Ludovico Biraghi Lossetti
President and chief executive officer
IBM *Italy*

During this period of worldwide inflation, Italy has had a more than ample share of economic problems. IBM Italy's president and chief executive officer, Ludovico Biraghi Lossetti, says his country is making "an all-out effort to meet the challenge of this difficult environment. IBM Italy can do much to help."

Biraghi Lossetti points to the skills of IBM people in bringing greater effi-

ing prime evening time.

Novedrate's daring architecture—initially a shock to some of the students—has since gained strong acceptance within and outside IBM.

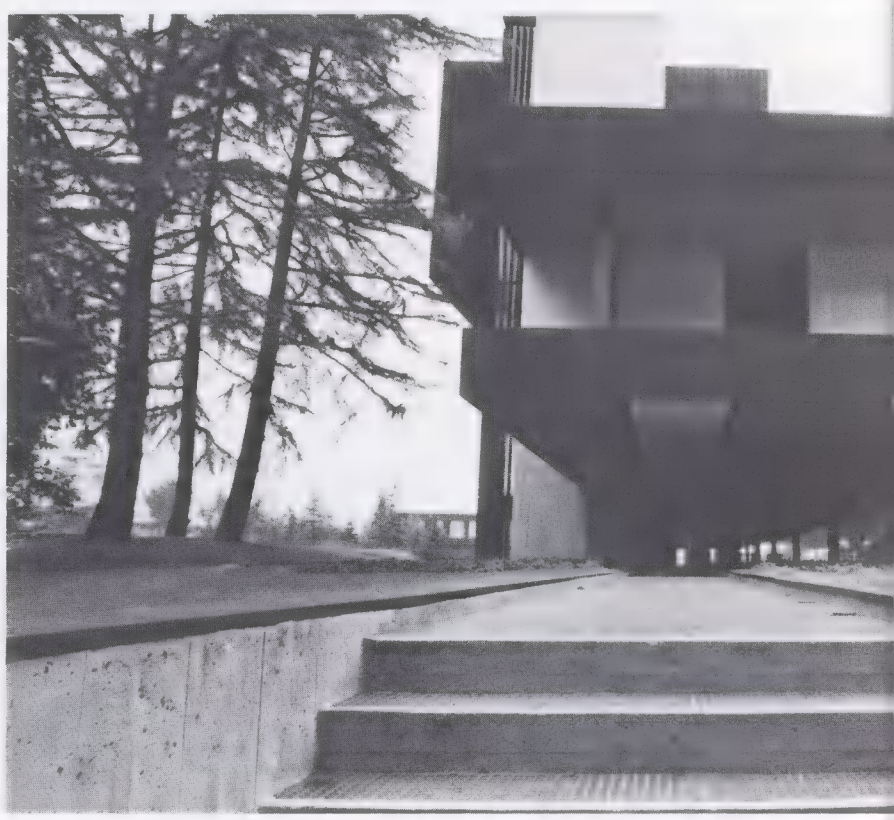
Says Luigi Sala, a systems engineer who has studied at the center: "At first, you're rather overwhelmed by its rather stark efficiency, its no-nonsense air. Then you begin to appreciate that it has a beauty of its own. It's comfortable, without being luxurious, and it's ideal for learning. To me, it's proof that IBM cares about the future."

Education centers worldwide

Novedrate is one of 47 national education centers in the IBM World Trade Europe/Middle East/Africa Corporation, which annually give courses to a total of more than 100,000 customers and IBMers. Among other major country sites are the Boulogne Center, just outside of Paris, and the Nordic Education Center in Lidingö, near Stockholm.

In addition, six international schools share facilities at IBM's new education center at La Hulpe, Belgium, outside of Brussels, which started operations last September. When completed, this center will be able to accommodate up to 400 resident students, including DP marketing managers, systems engineers, administration managers, customer engineering managers, plus IBM and customer executives. Next year, an estimated 7,000 people will attend classes there.

In IBM World Trade Americas/Far East Corporation, all 45 countries have facilities for education, with residential education centers at Amagi, outside of Tokyo; Cuernavaca, near Mexico City; and Gavea, a new center near Rio de Janeiro, designed to house 140 residential customers and IBM students when it opens next July. The A/FE countries conduct over a quarter of a million student days of customer education annually.



Above: Standing on a 45-acre parkland, the center takes the form of a series of blocks, built mainly of iron, reinforced concrete, and glass. The separate sections are closely linked and easily accessible.

Right: The glass-enclosed dining rooms can handle 500 people. Between classes and study in their room or the library, students can keep physically fit by using the gym, three tennis courts, two bowling greens, and the basketball/volleyball court. Most courses run for one or two weeks.

ciency to government and business through data processing. "This is of the utmost importance if we are to stem escalating costs and expenditures and keep Italian goods and services competitive on European markets."

IBM Italy's chief executive, who says he's confident that his country will overcome its economic crisis, has shared the fortunes of IBM Italy since 1946, first as a salesman and then in various managerial positions. He became president and chief executive in 1972, and this year was elected a director of IBM World Trade Europe/Middle East/Africa Corporation.

During the last 10 years, IBM Italy's gross income has increased sevenfold, and the number of employees has more than doubled, to its present level of more than 8,000 people. IBM Italy ranks 17th among Italian companies in size. Within the corporation, the country has been a leader in introducing new customers to data processing. It has been highly successful in manufacturing, marketing, and applying small systems, especially the various models of the System/3, which—with System/370 Models 115 and 125—are produced at the Vimercate plant and exported throughout Europe, Africa, and the Middle East. ■

Right: Students use the game rooms and lounges before and after class. All classrooms have closed-circuit television, linked to an audiovisual center, which produces educational videotapes. Classroom terminals are tied to Novedrate's computer center, which contains various models of System/370 and System/3.

Below: Villa Casana, an 18th-Century country mansion on the grounds of the education center, was restored by IBM Italy. It is occasionally used for seminars and management meetings.

Work style

Let's get rid of jargon

When language is used to obfuscate, pontificate, or impress the listener with the wit and wisdom of the speaker, it is like stumbling through a forest of thought hung with parasitic Spanish moss. Mossy jargon doesn't totally conceal the idea tree, but it disguises and blurs the meaning.

Does it sound impressive to "finalize an upward corrective adjustment in product lines?" Not when such moss is translated into what is really meant: *raise prices*. How much simpler to trim the phrase of excess fat, which is *not* to say, excise the adipose tissue.

—(James J. Kilpatrick
Nation's Business, July '74)

Know when you work best

Some people are "morning persons"; others, "night persons"; and still others (probably in the minority), "around the clock" persons. Daily cycles of productivity differ quite ■ bit. The trick is to know your periods of peak energy. Then you can save your toughest assignments for those periods, and schedule routine work for periods when your energy is on the wane. The first step is to keep a record of the time of day when you feel the peppiest, when you're most fatigued, when you're sharpest mentally, and when you can't concentrate at all. Then, adjust your schedule and workload accordingly.

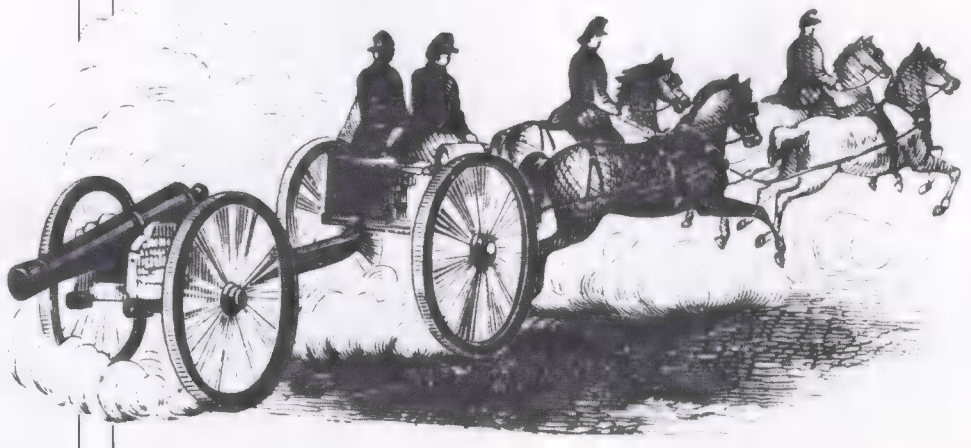
(*Keeping Young in Business,* Auren Uris)

Lincoln's solution

Sometimes, restructuring a problem provides a fresh insight—and a solution.

After the Battle of Gettysburg, for example, General Meade's army of 90,000 faced General Lee's army of 60,000. According to Meade's reasoning, he couldn't attack Lee because the odds weren't good enough: Two men on defense equaled three on offense. President Lincoln reviewed the stalemate, and then restructured the situation. If Meade did not attack, Lincoln explained, then he was just as much on the defense as Lee. Based on his 2-to-3 ratio for defense, Meade himself would need only 40,000 troops to defend Washington, D.C., against Lee's force of 60,000. That would release 50,000 of Meade's troops to harass the enemy elsewhere.

(*The Science Teacher,* April '74)



How much sleep?

Get whatever sleep you need to feel alert the next day, but no more, says Peter Hauri, director of Dartmouth Medical School's Sleep Clinic. The reason: Too much sleep can make you feel groggy.

Hauri maintains that there's no magic in the traditional eight-hour sleep formula. Most people average seven or seven and a half hours a night, but many function equally well on just five or six hours. An occasional sleepless night shouldn't impede your performance, according to Hauri, but two or three sleepless nights in a row might affect your work and health.

(*Business Week,* August 31, 1974, "How Much Sleep Do You Really Need?")

Playing it safe

To avoid friction and possible conflict, a decision-maker sometimes opts for the second best choice. The measure becomes one of what will be the most acceptable to the people affected by the decision. And this can lead to a "safe" or "let's not rock the boat" decision. The fallacy here is that the person making the decision does so, not on the basis of the facts and the logic that flows from them, but strictly on the basis of interpersonal relationships. The result short-changes both the decision-maker and the organization. It's too easy a way out.

(*Management Review,* August '74, "How Second Best Decisions Get Made")

Recycle yourself

From time to time, you may wonder if it's time for a change—in your expectations, your career, your life in general. If so, why not ask yourself the following:

- Is much of your life predetermined? Are you a bored spectator of your own existence?
- Are your own activities your main concern?
- Do you feel locked in by past circumstances?
- Have you achieved the goals of your youth, only to find the reality disappointing?

If your answer to most of these questions is "Yes," it's time to set new goals and begin to measure your progress. The very act of forming explicit plans—even short-term goals—is, in itself, an act of renewal and a major step toward accomplishing those goals.

(*Harper's,* September '74. "Wrap-around")

'Hurry' sickness

They call it the "high-risk personality." The person who has it is the doer-shaker-mover-achiever—hurried, impatient, tense, constantly under stress, and burdened with too many things to do in too short a time. This person is also a prime candidate for heart disease.

Any way out? Drs. Meyer Friedman and Ray Rosenman of Mount Zion Hospital's Harold Brunn Institute in San Francisco say an individual with these characteristics should restructure his or her work habits to cure what has become known as "hurry sickness"—possibly even save a life.

If you fit into this category, here's what these doctors advise: Recognize your limitations, take short breaks in your schedule to reduce tension, and allow time to read, think, and analyze.

(*Management Review,* April '74, "Stay Calm and Stay Alive")

PRIMER ON PROGRAMMING

by Gerald M. Weinberg
Professor, Human Sciences and Technology group, State University of New York, Binghamton

Several of my programmer friends recently had the idea of writing a children's book on programming. They want their children to understand what daddy and mommy do for a living, but they were having such a tough time explaining it that a book looked like the only solution.

I was struck by the irony that programming should be such an invisible, somewhat mysterious profession. For the word, "program," comes from Greek, *prographein,* "to set forth as a public notice." How have we evolved from public notice to the secrecy and mystery that seem to surround the act of computer programming?

Copyright © 1974 by Gerald M. Weinberg

Actually, programming has a long history. It is only *computer* programming that is new and unfamiliar.

Writing the program for the annual banquet of the Weeping Water Home and Garden Club, for example, *is setting down in advance* what will happen:

7:00 p.m.	*Prayer:*
	The Reverend W. Smith
7:05 p.m.	*Dinner:*
	(Sirloin Steak)
8:30 p.m.	*Speech:*
	"Roses Without Thorns"
	—Dr. L. Stookey
9:30 p.m.	*Question period*
10:00 p.m.	*Adjourn*

Though the program is a statement of what we would like to have happen, it doesn't always turn out that way. Dr. Stookey might decide to speak on "Thorns Without Roses," or he may drone on long after 9:30 p.m. *We* have

written the program, but other people may not follow it.

When someone fails to follow the *public* program, we usually invoke a *private* program, which got into our heads from some unknown past. We might just suffer silently through "Thorns Without Roses," vowing never again to invite Dr. Stookey; or if he talks too long, we may first cough suggestively, then finally stand up and walk out of the room. We could hardly write *that* into the program.

Whenever we program events involving human beings, then, although we may be victims of their changeability, we can also, being human, adapt ourselves to unforeseen changes. If people do not prove adaptable, the program must be worked out in greater detail. As an Olympic Committee member once remarked when complimented on the thorough programming of the games: "We must plan everything so thoroughly because we're not very flexible people."

Computers are not very flexible, either. And the detail required for computer programming is an arduous task involving much human trial and error.

In some ways, a programmer's experience with the computer is like the tale of the peasant and his wife who are granted three wishes. While they are debating the wisest possible wish, they grow hungry and the wife sighs: "I wish I had a sausage." Pow! A sausage appears on the table and one wish is gone. Furious, the peasant exclaims: "Foolish woman! You've wasted one of our precious wishes. I wish that sausage was hanging from the end of your nose." Pow!

Any programmer knows what the third wish had to be—unless the peasant wanted the sausage to stay where it was.

The programmer's problem with the computer, which carries out orders quite literally, is like the peasant's with only three wishes. He has to pay careful attention to particulars if he wants the right results. In common human public discourse, we are accustomed to specify rough programs and have others fill in the details.

■ Programming is foresight.

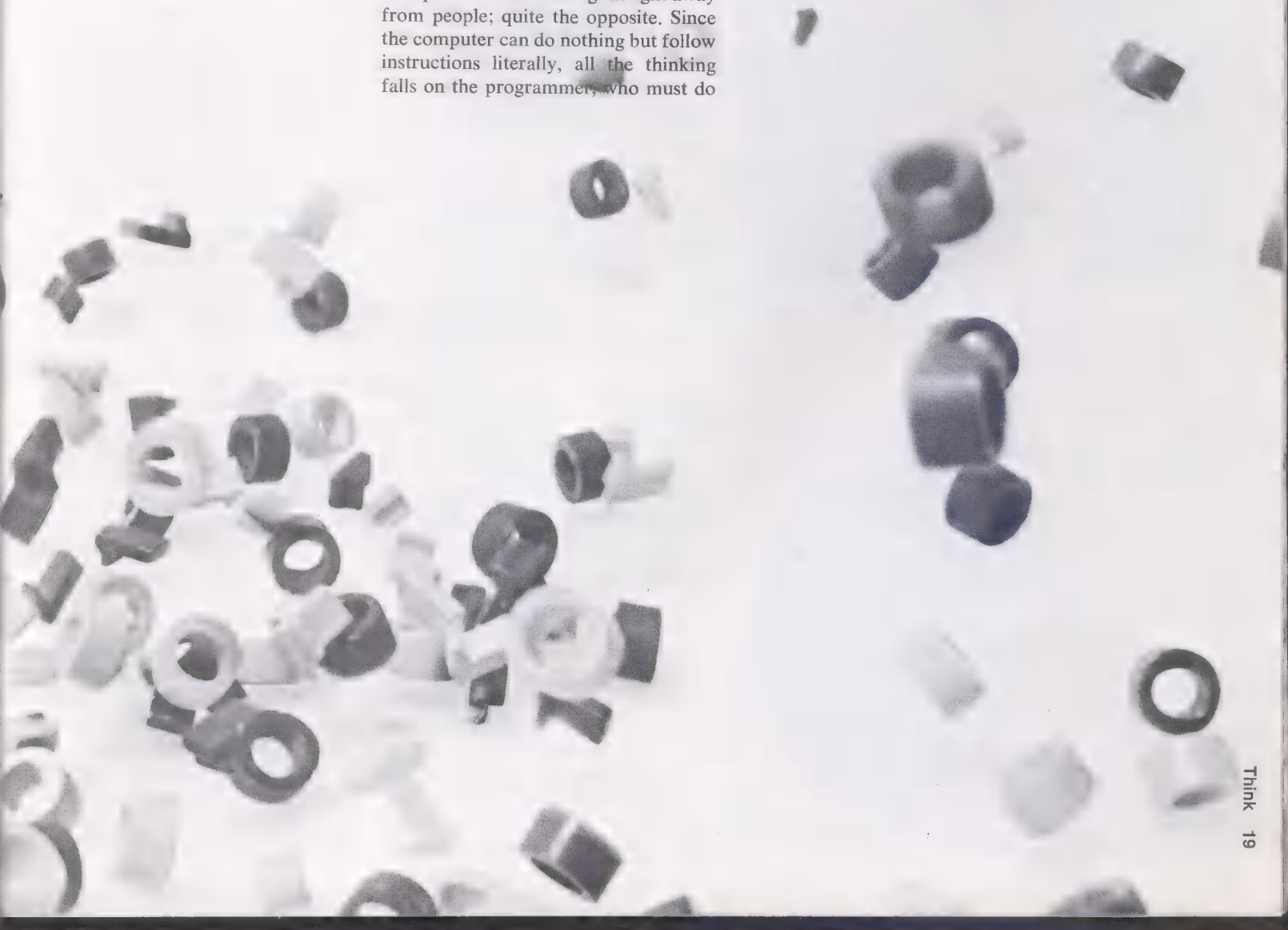
I consider it a great piece of foresight that IBM named its magazine *Think*. Computers are not taking thought away from people; quite the opposite. Since the computer can do nothing but follow instructions literally, all the thinking falls on the programmer, who must do this thinking *in advance*. And this makes programming difficult, for we are not accustomed to thinking in advance.

■ Programming is writing the recipe, not baking the cake.

When you bake a cake, whether it wins a prize or burns to a cinder, you can eat it only once. When you write a recipe, however, you are writing about events in the future, and many different cakes can be baked from the same recipe. How can this be? As any good cook knows, each recipe can be adjusted for variations in ingredients, idiosyncracies of the oven, or just plain little mistakes that happen along the way. But while the recipe writer can assume that a human baker will be able to adjust "6 large eggs" to "7 medium eggs" without being told, the computer programmer can assume no such thing.

■ The premise of programming, as of science, is: "If-so-then-so."

Both scientists and programmers talk about the future. The scientist says, in a law: "This will happen." The programmer says, in a program: "Make



this happen." But what the scientist really means is: "*If* that happens, *then* this will happen." Though science strives to make laws ever more universal, each law has a limited range of application.

Similarly, the programmer writes his commands to the computer in a contingency manner: "*If* that happens, *then* make this happen." If a programmer were writing a recipe for an automatic cake-baking computer, he might specify:

- If there are large eggs, then use 6 large eggs;
- Else if there are medium eggs, then use 7 medium eggs;
- Else if there are small eggs, then use 8 small eggs.

■ The programmer's recipe must be explicit.

Look again at the statement:

- If there are large eggs, then use 6 large eggs.

To us, it *looked* clear, but a programmer must recognize that it was supposed to say:

- If there are *at least* 6 large eggs, then use 6 large eggs.

Otherwise, the computer doesn't know what to do if there *are* large eggs, but not as many as 6. In such an ambiguous situation, the computer will always do *something*—but we don't know *what*. It might stop working on the cake, leaving a half-finished batter; or it might use the 2 large eggs it has and go on baking the cake, which would turn out dry and heavy. *Whatever* the computer chooses to do, it is unlikely to be anything we would call "sensible."

■ The programmer's recipe must be complete.

If we correct the recipe to remove that ambiguity, we face a further dilemma: What happens if there are not sufficient eggs of any size? Does the computer go shopping? Borrow from a neighboring computer? Stop in mid-recipe? As the program now reads, it will simply make the cake *without eggs!* The program is unambiguous, but *incomplete*.

■ The programmer's recipe must have an ending.

Suppose, to cope with the remaining situation, we add this line: "Else go to the store and buy more eggs." By this addition we generate new shopping situations:

- If they have at least 6 large eggs, buy 6 large;
- Else if they have at least 7 medium eggs, buy 7 medium;

Lady Lovelace

Her infatuation with the Babbage machines made her the world's first programmer.

Crown copyright. Reproduced with permission of Controller of Her Majesty's Stationery.

"Is thy face like thy mother's," wondered Lord Byron in his poem, *Childe Harold's Pilgrimage,* of his infant daughter Ada—"sole daughter of my house and heart?" As the painting shows, Augusta Ada, later the Countess of Lovelace, who is recognized today as the first programmer, bore a striking resemblance to the father she would never know. He left England for Italy in 1816, when she was a few months old, and died there when she was eight.

It was at his request that she became accomplished in Italian and the violin. At 15, she was also exhibiting an extraordinary precocity in mathematics. She liked nothing better, with the possible exceptions of dancing and horse racing, than to go with friends to Charles Babbage's workshop in London's Dorset Street for a look at the Difference Engine. There, noted one observer, "while the rest of the party gazed" with the expression of savages "on first seeing a looking glass . . . Miss Byron, young as she was, understood its workings and saw the great beauty of the invention."

At 27, with Babbage's encouragement (he remained a lifelong friend of both Lord and Lady Lovelace), she translated Luigi F. Menabrea's *Sketch of the Analytical Engine Invented by Charles Babbage,* which was based on lectures Babbage had given in Italy. Her copious mathematical notes, twice as long as the original paper, include detailed steps for the machine's operation, and display both literary and mathematical skill.

She turned her considerable mathematical skills to other pursuits as well. With her husband and Babbage, she tried to mastermind an infallible racing system. But, like most attempts of that kind, it didn't work, and she was hard pressed by debt before her untimely death, of cancer, in 1852.

Lord Byron

• Else if they have at least 8 small eggs, buy 8 small;
• Else go to another store to buy eggs.

Clearly, this *recipe* could be indefinitely long, but the programmer can shorten the *program* by using a "loop":

Repeat the following:
• Find a store not previously visited;
• Shop for eggs;
• *Until* sufficient eggs are found and purchased.

We could well imagine our computer shopping in vain at all the stores in Frankfurt and then catching a plane to Tokyo to "find a store not previously visited."

■ The program must be broad enough for wide use.

Our cake-baking computer has not been programmed to make substitutions. Any human baker knows that we could use 3 large and 4 small eggs, but the computer knows nothing unless told—by the programmer. What about 2 large, 3 medium, and 2 small? The more different contingencies there are, the more widely applicable the program—which helps spread the cost of programming.

■ The programmer's recipe must be verifiable.

The program's owners want assurance that it will perform well: that it will bake a cake, not a muffin or a brick; that it will leave the kitchen clean, not coated with six inches of sticky batter. As the number of contingencies grows —to make a widely applicable program —assurance becomes difficult to find. In a simple ten-step recipe with only ten contingencies per step and no loops, there would be 10 billion cakes to bake in order to try all possible combinations.

■ The programmer's recipe must be carefully controlled.

Since foolproof testing cannot be used to verify programs, the programmer must exercise *extreme control over the baking process.* For instance, we might express the quantity of eggs by *weight,* rather than by count: "Use between 340 and 380 grams of egg." By reducing many separate cases to a few, we get more assurance from fewer tests.

■ Large programs are mastered by successive refinement.

By deferring consideration of exactly *how* the egg will be obtained, the programmer can *start* writing the recipe in understandable terms. This starting level is the level of human thought. Much of the confusion and mystery surrounding programming stems from a failure to begin with such a humanized beginning, or from a failure to proceed by comprehendible steps. Each step must be done with exquisite care, lest the original meaning be lost in translation.

Using such a disciplined process, the programmer can write recipes of astounding complexity. Proceeding haphazardly, he can write recipes of astounding chaos.

■ The programmer has many tools, some of them programs.

It is not surprising that the programmer often turns to the computer itself for needed help. For instance, the computer may retain a "program library" or "programming language" to specify common tasks. Once one programmer has built a general method of obtaining between x and y grams of egg, future recipe-writers may work in a "higher-level" language.

■ Programming is more than communicating with computers.

When we try to write programs to make computers do things never done before, we inevitably discover that the future is not as simple as the past. New ideas and desires emerge from seeing our old ideas and desires implemented by the computer.

Because of new desires or old errors, programs must eventually be modified. When they are, someone must be able to understand them to make the modifications. Far more effort may be spent in modifying than in writing, so the programmer writes not just for the machine, but very much for other programmers.

■ Programming is a mirror of the mind.

The computer makes almost anything possible, like the peasant's wishes, if we but know *how* to ask. Programming, perhaps even more than politics, is indeed the "art of the possible." Before any programmer can translate wishes into instructions, people must understand their wishes, make them public, and balance them against the wishes of others.

Yet, as we attempt to program increasingly complex tasks, we realize—with the peasant—that we don't even know *what* to ask for. Because we programmers have heretofore been overly concerned with *how* rather than *what,* the art has seemed to be a private one concerned with machines. But it is, in fact, a public art concerned with reflecting the wishes of man. ■

Beyond the primer stage

A new programming series

In October, the first book of IBM's new Systems Programming Series is scheduled to appear in bookstores across the country. It's called *The Program Development Process—Volume 1: The Individual Programmer.* Its author is Joel Aron, manager of networking systems development, System Development Division, Gaithersburg, and editor in chief of the series. Aron defines the role of the programmer through the various stages of the system development life cycle—from concept, through design and coding, to testing, maintenance, and upgrading.

This winter and next spring, four more books are scheduled for publication. They deal with data base systems, programming languages, recursive programming techniques, and sorting systems. Their authors: C. J. Date and John Nicholls of the SDD lab in Hursley, England, and William Burge and Harold Lorin, staff members, respectively, of the Thomas J. Watson Research Center, Yorktown Heights, and IBM's Systems Research Institute in New York City.

There are half a dozen more books to come—on subjects which include mathematical background of programming, structured programming, compilers, and interactive computer graphics.

The series is a unique collaboration between IBM and the Addison-Wesley Publishing Company for the benefit of everybody involved in data processing, college students included.

State of the art

by Claire Stegmann

Some time ago, a harassed programming manager came across Sir Isaac Newton's famous quote: "If I have seen further it is by standing upon the shoulders of giants." On impulse, he sent it along to his boss with a note attached: "The trouble with programming is we all seem to be standing around at each other's feet."

The manager was reacting, like a concerned parent, to the frustrations involved in bringing to the unwieldy young world of computer programming some of the maturity already reached by its older professional sibling—engineering. The job isn't easy.

Reasons:

- Programming development is, as yet, an unpredictable child—slow at meeting its schedules and inclined to overspend its allowance.
- It is growing like Topsy. True, new

programming languages, control programs, and other "tools" have helped increase programmer productivity—there are an estimated 300,000 programmers in the U.S. today, barely double the number of seven years ago. But today's computers call for ten times as much programming as a decade ago.

- It is almost too popular. As new languages make user applications easier, computer users are calling for more, and better applications.

More than half the U.S. data processing budget—$24-billion is the 1974 estimate—goes, not for equipment, but for people, the majority of it in software. As programming costs keep going up ($10 per line is a recent estimate), hardware costs are coming down. (Memory costs are only a tiny fraction of what they were ten years ago.) Like the younger sibling of a brilliant elder, software is feeling the pressure to equal the strides that have been made in engineering.

At this year's National Computer Conference in May, George Glaser, president of the American Federation of Information Processing Societies, admonished the computer industry for being "machine mesmerized" to the point of not taking into account the growing inability of the programming profession to produce enough good software, speedily and cheaply enough to keep up with the advanced functions of the machines.

Everyone agreed there was a problem, but blamed someone else. Industry executives alleged that educators were producing poorly trained programmers; educators called for more cooperation from industry; and customers scolded programmers for tolerating too much laxity in the profession.

The urgency of the situation was reflected in the words of Professor E. W. Dijkstra, of the Technological University, in Eindhoven, the Netherlands. "What we are called upon to design," he said, "will become so complex that elegance [meaning scientific precision] will no longer be a luxury but a matter of life and death."

Dijkstra is one of three European mathematicians—the other two are Niklaus Wirth, of the Eidgenössische Technische Hochschule, in Zürich (he is an IBM consultant), and C.A.R. Hoare, of Queens University of Belfast, United Kingdom—who, with others, have been pursuing ideas toward better education in program design.

In the U.S., working independently, an IBM Fellow, Dr. Harlan Mills (*see page 24*) has come to similar conclusions.

The new techniques—structured programming, chief programmer teams, and topdown development—were used with spectacular success (*Think*, August 1973) in *The New York Times* data retrieval system, under contract with IBM's Federal Systems Division.

Says FSD President John B. Jackson: "Programming has always been a significant part of FSD's business. Today it is increasingly so as the division seeks a larger role as a systems integrator. Programming requirements for such systems are always demanding, either because of the complexities of the functions or because of the environment in which we operate. Whether in the national defense or a vital space program, the margin for error does not exist."

The new programming techniques are being used in IBM's System Development Division, along with others with names like "walkthrough" (a review of each other's work by a cross section of programming planners, publication writers, developers, and testers, including product assurance support); HIPO (an acronym for Hierarchy plus Input, Process, Output), which is a new and better form of flow chart; and cause-and-effect graphs, which help to illustrate the Boolean logic on which programming is based. (George Boole,

---

Claire Stegmann, senior editor on the staff of Think, *provided editorial direction in the preparation of this Special Report on programming.*

Discipline

That's something good writing and good programs have in common, says Harlan Mills. He has led in applying it to the art of programming.

"I tell my students," says IBM Fellow Dr. Harlan Mills, who also teaches graduate courses at The Johns Hopkins University, "that to become good programmers the best book they can read is Strunk and White's *Elements of Style*.

He is referring to the generally recognized "bible" of the writing profession, a spare little volume of 78 pages, now in its 2nd edition, which includes all the rules and principles for writing good English anyone need know.

In his way, Dr. Mills—through "structured programming," "chief programmer teams," and "topdown development"—has provided much the same service for programmers.

It takes at least as much work to put together a complicated program as it does to write a good novel, Dr. Mills says. But in the past, many programs have been rambling streams of consciousness, jumbles of thought and afterthought, so complicated that even their authors, returning to them months later, have found them unreadable. Many programmers wrote complicated programs simply because they didn't know there was a simpler way. Seven years ago, Dr. Mills, then a mathematics consultant to the Federal Systems Division, decided there must be. He studied programming to find out.

"Mathematics usually concerns itself with physical things," he says, "but we found a mathematics to fit the mental process of programming."

Dr. Mills attacked both the physical form of a program and the thinking behind it. His structured programs are programs which are divided into self-contained units or "pages" which use a series of indentations and three kinds of control logic, "nested" within each other like Chinese puzzle boxes, to do for programs what sentences, paragraphs, pages, and chapters do for books—namely, make them easier to read (and, incidentally, to debug).

Topdown development, on the other hand, is to a program what good outlining is to a book. The author, or chief programmer, first determines what it is he or she is trying to say, or, in the case of a program, do.

To illustrate with a simple problem for the computer: "Given a set of numbers, find the largest and the smallest. Take the smallest from the largest and print out the results on a terminal."

Fifteen years ago, a programmer might have started the program by reading in the first number. With topdown, he starts by thinking through, and writing down the procedure for solving the problem, in its overall terms.

What are the things to be done?

1. Read the numbers.
2. Search for largest and smallest.
3. Compute largest minus smallest.
4. Print out the result.

At this point, the programmer is not concerned with details of computer memory or input devices. The whole process is described on one page. Then, each subprocess is described on a new page, and so on.

As the program design unfolds, the chief programmer assigns a programmer or, in a large project, a whole chief programming team, to continue major subprocesses to (in our example) each of the four steps. They, in turn, write out their portion of the problem in general terms, a page at a time. Again, a programmer is assigned to each step of the work plan, and so on down the line.

The operation expands from a base with four branches into what is known in mathematics as a tree, with many leaves. The leaves represent the blocks of programming (again in page form) to be written. When they are strung out and combined, they form the program.

Since each group works independently, the chief programmer of each team has responsibility to interrelate with other groups where necessary.

If all this seems familiar to managers, it is not surprising. Says Dr. Mills: "What topdown development, or design, gets down to finally is isolating subproblems and delegating authority. That's something good managers do all the time."

a 19th-Century English mathematician, showed that the solution to even the most complex propositions can be found by deduction, using only two symbols, to represent "either" and "or"). All the new methods share a common goal—to divide a task of gargantuan proportions into work blocks which can be more easily managed and measured.

Computer experts who were around 20 years ago, when it all began, believe the new techniques represent a much-needed return to discipline. Says one: "In the early days we spent a great deal of time getting a job thought through because none of us really knew what programming was. We didn't have the bag of tricks we have today. Over the years, however, the whole field has gotten into some very sloppy habits."

Programming a computer of the 1950s or 1960s is a far cry from the jigsaw complexity of modern data processing in use today. Within a time span of only two or three years, hundreds of programming pieces, often built in scattered development locations, must now be brought together, sent back for further tailoring, tested and retested, to ensure a harmonious fit for today's giant systems.

Keeping up with what is happening in the field has become a formidable task. Says Ted Climis, SDD vice president, system development: "I used to look forward to the time when I would be 35 and would know all there was to know about programming. Well, now I'm over 45, and I'm still waiting for that day. The amount of self-education you need just to keep going is unbelievable.

"It's not like a tidal wave; it's more like a sandstorm. Because a tidal wave will pass over you, and either you'll survive or you won't. But a sandstorm will bury you, inch by inch. Your feet get weighted down. You know you've got to run like the very devil just to stay even, and there's no way to stop the sand."

Because programming, unlike engi-

neering, cannot afford the luxury of a pilot line, it is a kingdom that can be lost for want of a nail. A single programmer, making the wrong tradeoffs and failing to write one piece of code during development has been known to cause a whole schedule to slide when the time comes for final assembly.

In the five years since IBM began separate pricing for its program support, it has learned that heavy use of chalkboards, telephones, and face-to-face contact in early design phases can greatly reduce costs at the end of the programming cycle, when errors become problems for customer engineers.

"When I joined IBM years ago," says George Radin, manager of advanced systems architecture, SDD, who was a key developer of the programming language PL/1, "the skills required were those of the garage inventor, carefully constructing a better mousetrap. But gradually one-man jobs have become jobs involving 200 or 300 people.

"When I hire people for programming these days I'm in some sense as much interested in a person's ability to work with others as I am in technical skills. More and more, programmers, whether part of management or not, are apt to be "committee" members, presentation-makers, persuaders, explainers, teachers, debaters. Programming is not something you do in isolation anymore."

IBM is building other lessons it has learned directly into its own development support system. "We are using our own systems, and building on those systems," explains T. R. Hunter, director of programming development, processors, SDD Endicott, "those things we think are required to control the process—checkpoints on productivity, error rate, cost estimate against budget—at various stages of the programming cycle. The time has come to apply some of the yardsticks we have developed for customers, in the form of management information systems, to our own program development business." ■

Team portrait: DPD's first chief programmer team includes, left to right: Gail Agnew, functional analyst; Mike Hoodes, chief programmer; Bill Royall, program manager; Gloria Garafolo, librarian; and Frank Fandrick, backup chief.

Portrait of a programming team

Two years ago, Bill Royall, program manager in the Data Processing Division's health industry development department, made programmer Mike Hoodes an offer he couldn't refuse. Royall wanted Hoodes to head up the first chief programmer team in the Data Processing Division.

Hoodes had misgivings. "The way I understood the role," recalls Hoodes, "there were maybe three people in the country who could fit the job."

A chief programmer leads a team consisting of a backup programmer, who acts as his stand-in; a functional analyst, that is, a programmer with experience in the particular field the program is to be written for; and a program support librarian. Ideally, the chief has a great deal of in-depth experience in programming planning and development and holds an advanced degree in math or computer science.

True, Hoodes was a former math teacher. And at IBM, he had quickly acquired the nickname, "Macro Mike," for his habit of coming up with new, easier ways of doing things. (A single macro instruction is interpreted by a computer as a whole block of instructions, and saves programming time.)

"Mike and Bill have what I call intellectual honesty," says Steve Lucas, DPD health industry development manager. "They're both willing to try something different when they see the need."

The need was obvious. The health group of some 30 IBMers who develop programs primarily for hospital administration, was spending more than half its programming time searching out and correcting bugs in existing programs, leaving little time for new development. Hoodes agreed to do his best, and went off to chief programming school at the Federal Systems Division in Gaithersburg.

Chosen as his backup was Frank Fandrick, a certified Wisconsin cheesemaker (he keeps his U.S. Department of Agriculture certificate on his chalkboard), who learned programming in the Air Force. Gail Agnew, a former medical researcher, agreed to act as liaison between the team and the applications area, a medical laboratory. The job of support librarian went to Gloria Garafolo, a crackerjack secretary, but with no programming experience.

The team spent a week learning structured programming. Then everyone, Royall included, went through a two-day team building session to learn togetherness. "It was rough," says Royall, "but it paid off. We really began to level with each other."

During their 24 months together, they were able to measure performance on a weekly basis. Programmer productivity went up 36 percent, in terms of lines coded and tested. Librarian Garafolo saved each programmer two hours a day by centralizing housekeeping tasks through her terminal.

Gail Agnew, sharing Hoodes's original misgivings, was converted to structured programming one day when she found herself alone with the responsibility to make a major program change. To do so, she needed to be able to interpret code written by other team members. She did it in a couple of hours; similar changes, in the past, had required three days or more. "I'd never go back now," she says.

Neither would Hoodes. Last August, he left for Europe, to spread the word to customers abroad as an advisory systems engineer to the IBM Europe/Middle East/Africa Corporation.

People in programming

Application developer

"We design programs for small businesses the way tailors make clothes—to fit individual needs."

That's how Terry Brown describes his job as project manager in application development at the General Systems Division in Menlo Park, Calif.

Each customer specifies what he wants by filling out questionnaires and, in return, receives a system application made up of a language of the simplest programming terms, called the Report Program Generator.

On the basis of the questionnaire forms, GSD selects pieces of computer programs from its application library, stored in a host machine at Menlo Park. The largest industry design in its library is BICARSA, which stands for Billing Inventory Control Accounts Receivable Sales Analysis.

"It saves development time for us," says Brown, "but still gives the customer flexibility of choice."

Brown, a Drake University graduate in actuarial science, is a 33-year-old motorcycle buff who learned about customers during his four years as an IBM systems engineer in Iowa, working on both small and large accounts.

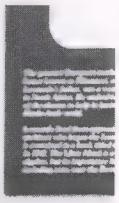
Chief engineer

Mrs. Mariko Kamijo, a senior systems engineer in an IBM Japan data center in Tokyo, is helping to solve some of the world's biggest traffic jams. She is in charge of the design of a computer application called Tokyo Expressway Enterprise Management Systems (TEEMS), which is being used to build Tokyo's giant expressway.

Japan has more motor vehicles—25 million—than any other country outside the U.S., much of it concentrated in and around its densely populated capital city. A dozen years ago, the Tokyo Expressway Enterprise began construction of the road network to crisscross the city and improve traffic flow. Today, some 527,000 vehicles daily pour over its 66.8 miles (107.8 kilometers). It is still growing.

Mrs. Kamijo, who is 35, has been working with the expressway since 1969. She authored Auto-Net, a program which automatically "plots" modifications of the original road system design. It can even "read" Chinese characters used in the Japanese language. Her group's current project is to standardize the construction methods used by the expressway's hundreds of contractors. If this can be done, she says, it will cut costs and improve system integration.

She is the only woman among IBM Japan's 25 Japanese senior systems engineers, a fact she attributes not to discrimination, but to a shortage of day care centers for the children of working mothers in Tokyo. Fortunately, she and her husband, who also works with computers, for the government-owned Japan Information Processing Development Center, found one for their small daughter.



0,1,1,2

Inventory tracker

Ferdinand Harig, a senior associate programmer from IBM Germany now on loan to the System Products Division in Sterling Forest, is part of a team working on a unique IBM program. It will enable terminals in SPD, the company's largest division, to "talk" to terminals in the company's overseas operations.

They use separate data bases. Both track inventories, but under different ground rules. Says Harig: "We're building a bridge between the two." (One is Common Manufacturing Information System—CMIS; the other is Information Management System—IMS.)

Harig, 30, got his training in computers as an apprentice technician in a German automated steel rolling mill. He joined IBM as a technician in Sindelfingen nine years ago, and when he returns to Mainz early next year, he will work in IBM Germany's central teleprocessing operation center. His job there: answering questions and helping to solve problems for fellow programmers and systems analysts.

They number more than 16,000 in IBM worldwide. Many have advanced degrees. About one-third have no degree at all. But, according to a recent Data Processing Division survey, of its own, they do have one thing in common—a passion for science fiction.

They're known as programmers. Nobody knows what makes a good one, although 70 U.S. universities offer Ph.D.'s in computer science, each with a healthy content of programming training. Experience shows that he or she should have a logical turn of mind and feel at home with symbols. Otherwise, programmers constitute a diverse collection of people, performing a rich variety of tasks, as the portraits below reveal.

Design manager

Pat Kearney spends most of her time meeting with programmers, monitoring specifications, and negotiating programming requirements with the Data Processing and Field Engineering Divisions and IBM facilities around the world. She's a manager in the System Development Division design office in Poughkeepsie. Speaking of her role as negotiator, she quips: "I think we could learn a lot from Kissinger."

She was functional designer for VS2 telecommunications access methods before becoming manager of the design control group for VS2.

A political science major in college, she joined IBM in 1969. These days she's doing more managing than programming. But she works hard at keeping up what she considers the key to good programming—creativity. And she thinks all design programmers should, by whatever means.

That's why she took up photography two years ago. "When you are accustomed to looking at things one way," she says, "learning how to use a camera effectively gives you many different perspectives."

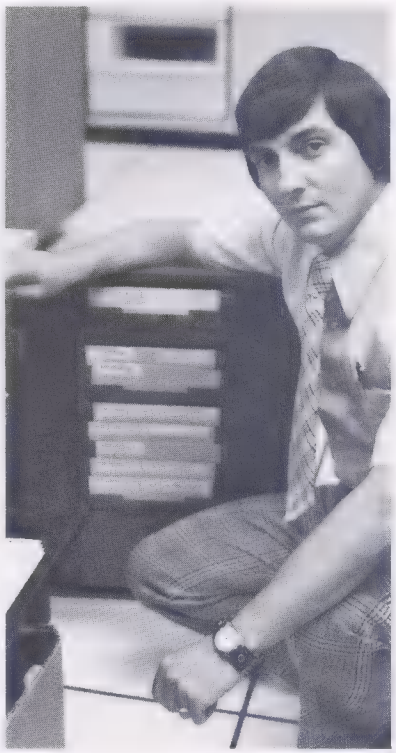
Micro-programmer

To perform a particular function, computer circuits can be either completely wired to perform that specific function or wired in such a way that they accept control signals generated from basic machine instructions called *microcode*. Both methods accomplish the same result. Complete wiring (hardware) is faster, but microcode gives users more options, allowing them the flexibility of special features such as emulators (a combination of hardware and microprogramming which allows one system to accept programs from another).

Art Nolta, a microprogrammer in the System Products Division in Endicott, explains: "Once a function has been wired in a system, it cannot be changed without rewiring. If the function is implemented in microcode it can be easily modified by rewriting the microcode. The microcode can then be transferred to a 'floppy disk' and easily inserted into the system.

"In other words," says Nolta, an avid hockey fan who finds microprogramming intriguing, "the functions performed by a microprogrammed system can be modified or replaced easily by reloading control storage with a different set of microcoded functions."

A graduate of the RCA Technical Institute in Cherry Hill, N.J., Nolta joined IBM 10 years ago as a test technician in Poughkeepsie. He has been involved in microprogramming, at Endicott, for the past two years.



The only language the modern digital computer directly responds to—binary code—is more than 400 years old. Long before Charles Babbage anticipated computer concepts with his Analytical Engine, Sir Francis Bacon, 16th-Century English philosopher and man of letters, was using a form of the code—which is based on only two symbols—to write secret messages.

In the 1940s, Dr. John von Neumann and IBM Fellow Dr. Herman Goldstine chose binary code to instruct the early machines, because of the essentially binary character of the computer's electronic components. A vacuum tube can, after all, exist reliably in only one of two states: on or off. A switch is open or shut; a pulse, either present or absent.

According to a fixed set of rules, the binary digits 0 and 1 (bits) can be combined to represent both decimal numbers and all the letters of the alphabet. In binary, for example, the decimal number *24* is expressed as *00011000*. A capital *J* is *11010001*, and a small *n* is represented as *10010101*.

Easy for the machine; but since every instruction must

Programming languages

They make computers easier to use

"Programming languages have become the major means of communication between the person with a problem and the digital computer used to help solve it," writes Jean E. Sammet in the introduction to her book, Programming Languages: History and Fundamentals. *In this interview with* Think, *Miss Sammet, who played a key role in the development of* COBOL, *led the development of* FORMAC, *and is now programming language technology manager in the Federal Systems Division, gives her personal perspective on questions frequently asked about the development of programming languages.*

Where does the pressure for the development of a major new programming language come from?

From every different kind of organization—users, computer manufacturers, research and development outfits. In the case of COBOL, it came from the Defense Department. Dartmouth College developed BASIC, because it wanted something simple for students to use. JOSS, the first of the on-line languages, came from the Rand Corporation.

Do users and user-groups play a fairly large part in the actual design of major programming languages?

Users could and should play a stronger role, but historically they have not. PL/1 is probably the best example in which the user-group was involved. There was a so-called "three-by-three committee," with three people from IBM and three people from the SHARE user-group which developed the language. However, after the initial work was done, then to a significant extent IBM took the lead.

By and large, most users don't have the technical knowledge that is needed. It's not as simple as you might think to sit down and design a programming language. One needs to know a lot about language design in order to even fit the pieces together, let alone make

be carefully spelled out, writing in machine language can be a long and laborious task for the programmer.

For this reason, programming languages have sprung up. They are written in a form more closely resembling the programmer's own working vocabulary. FORTRAN is an example in mathematics while COBOL can be used for business applications. The computer then uses a program called a compiler to rapidly translate these languages into binary code.

Almost all the programming languages, e.g., COBOL and PL/1, include many English words and phrases such as PERFORM, READ INTO. But the rules for these artificial languages are much more exacting than those for natural languages. The omission of a hyphen, for example, has been known to cause a rocket to explode.

The acronyms for the 170 computer programming languages used in the data processing field betray the human quality of the developers. They bear such names as SNOBOL, LISP, and MADCAP. Only about ten are of major significance. Among them: FORTRAN, COBOL, and PL/1.

sure the language meets its objectives. The language designer also has to know a fair amount about techniques for translating to machine language. Otherwise he or she might create a language which can't be effectively translated.

A language such as COBOL or FORTRAN is eventually adopted in the form of an industry standard so that users are not confronted with variations of the same thing. Does the process of creating standards and having to adhere to them foster or put a brake on language development?

Standardization does both. If you standardize very early in the development of a language, as we did with COBOL, you are freezing into concrete something with which you have had very little experience. There hasn't been enough time to determine whether what was valid on paper is valid in use.

On the other hand, if you delay standardization, as happened with FORTRAN, you improve the technical result. People will have had enough experience in the language to know what features are good and what features are bad. But everyone will also have acquired a vested interest in doing something that's a little bit different. And it is very difficult to try to pull everything together to get a single standard.

Like yourself, a number of outstanding people in IBM—John Backus, Ken Iverson, George Radin—have played an important role in language development. How much does any single individual really contribute in shaping a language? Or is it mainly a group activity?

Individuals can and do have a very strong influence. But in almost every case, languages have been too complex for a single person ever to define the entire language. I'm talking about the definition, not the implementation, which involves many people.

Do you take individual professional needs into account when you design a language? For instance, doctors often do things differently from lawyers or accountants. They all use different terms.

Approximately half the 170 computer languages in the United States today are for very specialized application areas such as computer-assisted instruction, graphics, simulation, equipment checkout, systems programming, and civil engineering—which has many languages. The moment you get into one of these languages, you immediately take into account a number of individual idiosyncrasies because you are using a vocabulary and vocational style that might be appropriate to the civil engineer, for example, but not necessarily to the electrical engineer. So you have both jargon and style to take into account.

In fact, this concern with different needs is, to a very large extent, the reason why we have so many programming languages.

What causes one language to be more widely used or more significant than another?

Practicality—meaning the language is suitable for a significant, although not necessarily large, class of problems, and good compilers can be written for it.

But, a number of other factors can be involved. The personal prestige and leadership of people who develop a language play an enormous role.

Then, too, some languages clearly create a spark comparable to the political charisma which often affects election results. This can cause a language to become popular—sometimes to a level of fanaticism. ALGOL and APL/360 have this quality.

Is there such a thing as a company position when you are developing a programming language?

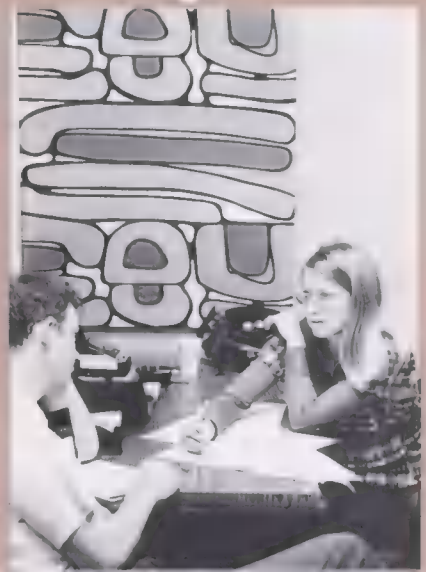
The assumption that many people have on points of language design is that there is an 11th Commandment which makes it obvious that this is the right way for the company to go. In my opinion, that's just not true.

On the original COBOL committee, for example, there were, in many cases, two representatives from the same company who fought more between themselves on a particular point than they did with people from a different company. When one is concerned only with the *technical* elements of a language, it is almost meaningless to talk about a company position. The major elements affecting a company position are marketing and economic considerations.

Will specialized application languages eventually be absorbed into larger generalized languages?

There's just no evidence that they will—and a great deal of evidence to the contrary. Personally, I think it's a very good thing that the specialized languages remain separate. When you give people a language that is very natural for them to use, they're going to use it more, and they are going to use the computer a lot more effectively. Some experts are known to disagree with this view. ■

Visit to a programming



center

The System Development Division's New England Programming Center is situated just off Middlesex Turnpike, near Burlington, Mass., a country rich in American Revolutionary history. As you turn toward the Center, on the edge of the New England Executive Park, you can almost see, in the fields to the south, a doughty band of colonial farmers rallying for a call to arms.

The Center, too, deals in illusion, of a history-making sort. It is the home of the care and development of SDD's Virtual Machine Facility/370 (VM/370). Here, in a one-story, brick-and-glass building, 115 IBMers—some just out of programming school, others 20 years with the company—practice the modern alchemy of turning one computer into many, all at the touch of a terminal.

VM/370 is an IBM system control program, which acts as a sort of master sergeant within a computer, scheduling and supervising the myriad tasks being processed, and assuring that jobs get done in as orderly and rapid a manner as possible.

Of the four system control programs announced for System/370 in 1972, three offer the advantage of virtual storage to small, medium, and large models, respectively, giving the effect of an almost limitless amount of memory space. The fourth, VM/370, goes even further.

With virtual machines (or VM)—which can be used alone or with the other system control programs—the user, through terminals, creates the effect of not only more main storage, but of many different-sized computers, complete with various kinds and quantities of simulated card readers, disk files, line printers, and other input/output devices.

Put another way, VM does for business operations what a car might do for family life if it could suddenly be transformed, like Cinderella's pumpkin, into not only a station wagon, but a dune buggy, a truck, a luxury limousine, or any other vehicle required by conflicting family needs.

The concept grew out of a company need to measure the efficiencies of various combinations of software and hardware in a time-sharing environment. Since VM/370 was offered to customers in 1972, it has been gathering ground, according to industry analyst David Eisenhauer, like a snowball rolling downhill.

"What happens on the 'real' 370," he explains, "is that the virtual machines are multiprogrammed—each task is allowed a slice of time, from a millisecond to perhaps seconds."

As terminal operators get answers, within seconds, with the computer, from terminal keyboards sometimes thousands of miles away, their instructions are invoking "pages" of programs on an extremely speedy direct access storage device, which moves them in and out of main storage as required.

After processing, if many terminal operators are using simulated printers as output devices, and there is only one real printer available, VM/370 simply "spools" the printouts onto an auxiliary storage device, or file, and queues them up for such time as the real printer is free.

When a number of terminals are used—a single System/370 can accommodate over a hundred—there is a tradeoff to be made between speed and flexibility; a computer may take longer to respond. But the difference in speed between the terminal and the more conventional method of data processing is like the difference between the automobile and the horse.

In the old days, Eisenhauer explains, a programmer had no choice but to hand a program to a computer operator or take it personally to the machine room. There it was keypunched onto cards, run through a card reader, and, in the unlikely event that the computer was not in use, printed out. In case of error, the whole process might have to be repeated. "Turnaround, or response time, on that kind of thing is pretty good if it's twice a day," he says.

"Now programmers no longer have to walk to the computer room. All their work can be done while sitting at the terminal. They can do what they want to do when they want to do it."

Eisenhauer, a Kansan, who is distantly related to the Eisenhower side of the Kansas family, has a math degree out of Washburn University in Topeka. He began his IBM career eight years ago as a systems engineer, and worked briefly as a DPD marketing rep before being involved in early VM development.

He maintains close contact with DPD and attends meetings of the customer groups, SHARE, GUIDE, and SEAS (the European chapter of SHARE). Such groups submit resolutions every six months to IBM, stating what they would like to see in IBM products.

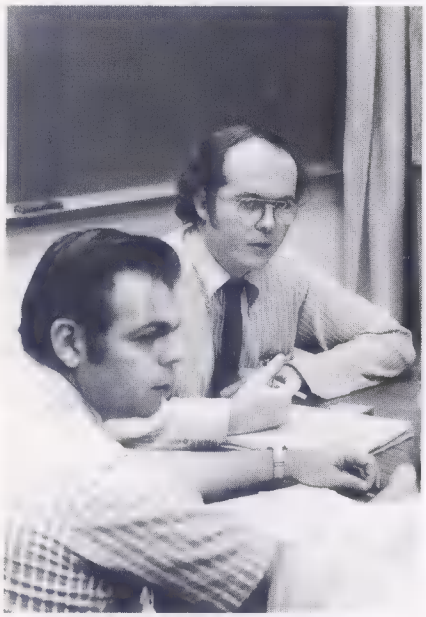
"I've challenged many a user," he says, not without pride. "Give me all the 'wouldn't it be nice if' ideas you've always had about data processing, and one way or another we can make them happen with VM."

■ "The Bitterness of Poor Quality Lingers Long After the Sweetness of Meeting Schedules," declaims the hand-lettering on the wall above Charlie Weagle's 3270 terminal.

Thump, thump, thump go the blue keys beneath Weagle's fingers.

Weagle, a staff programmer in the development department, is demonstrating, with the 3277 screen support he co-developed for the 3270 terminal,

The New England Programming Center at Burlington does the planning, development, and testing for one of the four system control programs for the 370, and keeps in touch with other SDD programming locations worldwide. At left, Sharon Koblinsky, control program development manager, discusses a point with Charlie Johnson, senior associate programmer.



Programmers meet regularly to track every stage of a VM/370 feature, from design through final tests. This team is led by product administrator, John Conway, right, and includes staff development programmer Charlie Weagle.

Staff programmer Dick Jensen has the job of testing VM/370 functions. His tests are often twice as long as the programs he tests.

how VM can be used in development work.

"Now, suppose I say, 'Let me have a disk with five cylinders.' The system will search around and say, 'Yeah, here's a disk with five cylinders, give it to Weagle.' " At his touch, within seconds, small yellow figures, indicating the computer disk address, appear on the screen above his terminal.

Weagle is as much at home in both software and hardware as he is in his native Massachusetts. A member of IBM's first System/360 Model 30 school for customer engineers, he was good at hardware and was not too happy when his manager gave him a programming assignment. By the time System/370 came out though, "I was up in Waltham shooting bugs [checking programs for errors] full time, and I liked it. So when I heard about this group, I joined up.

"I don't write programs," he explains, "I write functions." (That is, programming to direct some action a machine takes, like the return of a typewriter carriage.)

His toughest assignment came last year—to develop the graphic (screen) support for the Model 168. The Center has a Model 145 and a 155, but no 168. Weagle used VM/370 to simulate what he needed on his 3277 screen. When test time came around he took his work to a real 168 in Poughkeepsie. "It had a couple of bugs. I patched them up, and it worked fine."

■ Sharon Scheibel, a programmer of five years' standing, knows about bugs. She works in the Center's problem determination group, which processes APARS from the field. An APAR (Authorized Program Analysis Report), she explains, is a record sheet made up on each bug reported by a customer, or a customer engineer.

The problem determination group, also called, appropriately enough, a "change team," diagnoses and solves these problems. "Some change teams work on one small section of a system," explains Dom LaCava, manager of the group, "but we're general diagnosticians for everything. We're sort of like CEs from the field, only in-house."

At times, if a customer has a severe problem, members of the change team will travel to the customer's location and work with the CEs to solve the problem.

LaCava has won two Outstanding Contribution Awards for his work in the support of VM/370. He learned programming as a computer operator in Cambridge. "SEs would come in with customers," he recalls, "to benchmark their systems at night. In other words, they would run the system against a standard, to make sure that it measured up. There's a lot of baby-sitting with benchmarking, and at 3 or 4 in the morning we would sit and shoot the breeze. The SEs answered a lot of questions for me."

■ "Fortunately my wife is understanding," laughs Dick Jensen, "because once I get going on a problem I'm with it night and day until it's done. I even dream about it. I solve an awful lot of problems in my sleep."

Jensen is a staff programmer in the Center's systems test department. He's often behind a stack of VM/370 program listings, computer printouts of program contents. System control programs are put through an extensive series of tests. They also contain test programs by which the computer tests itself. Jensen writes such programs, as well as programs to test "in every conceivable way" VM/370's ability to simulate the real System/370. His test programs are often twice as long as the programs or functions he tests.

Unlike Weagle, who spends much of his time at his terminal, Jensen spends much of his time directly with the computer. Fairly sophisticated machine maneuvers can take two weeks to test sufficiently. "Sometimes," he says, "you have to do a lot of research to get a handle on what's happening."

In his seven years with the company, Jensen, who joined IBM from high school, and was a senior program systems representative on the Boeing account in Seattle before coming East, has spent more than two years in IBM schools. He keeps learning on his own.

"Right now I'm studying new functions. The more I study and learn, the easier it is to write comprehensive test programs. A lot of thought and concentration is required. It's not the type of job where you can come to work, put in eight hours, and go home. I don't want to anyway. It's too exciting." ■

outlook

The need is to develop programming that will make the computer a do-it-yourself tool

by Richard Bode

In the opening chapter of his book, *Design of Man-Computer Dialogues,* author James Martin, a member of the IBM Systems Research Institute in New York City, makes the following observation about the task that confronts programmers in the days ahead: "Increasingly in the next decade, *man* must become the prime focus of system design. The computer is there to serve him, to obtain information for him, and to help him do his job. *The ease with which he communicates with it will determine the extent to which he uses it.* [Italics ours.] Whether or not he uses it powerfully will depend upon the man-machine language available to him and how well he is able to understand it."

The statement is one with which IBM executives readily agree. "If we want to understand future programming requirements," says Dean McKay, IBM senior vice president and group executive, DP Marketing Group, "then we must look at the needs of the marketplace. And when we look at the marketplace, we see our customers want exactly what we want. Our customers want us to make programming easier so that more of their people can handle more jobs at lower costs.

"For every dollar the customer spends on IBM equipment," McKay adds, "he spends two dollars on his own people and data processing costs. And he spends another 35 dollars on administrative chores the computer hasn't even touched yet. So there's a vast, untapped market potential out there."

McKay observes that right now two-thirds of the money the customer spends on programming goes into maintaining the jobs already computerized. That leaves only a small part of the total programming budget for new applications. This fact alone puts a tremendous burden on system design and software development. McKay feels manufacturers must give customers the tools they need to more easily write application programs tailored to their own requirements.

"Developments like the application customizer for System/3 make an excellent start," he says. "But we must do more—especially with our larger, computers—to simplify the whole complex task of program preparation. At the same time we must remember that past developments in both hardware and software have dramatically increased speed and storage capacity. Inevitably, these advances extended the use of the computer to many people who had no experience with it before."

If anything, the number of new and inexperienced users is going to proliferate in the days immediately ahead. John F. Akers, president of the Data Processing Division, believes the new family of IBM terminals (introduced in September) will encourage many customers to install teleprocessing networks. The new 3767 desk-top communication terminal could be used, for example, by an employee at a service counter to respond instantly to customer inquiries, or by a store clerk in a sales department to expedite customer transactions. "Ideally," Akers says, "we

---

Richard Bode, a freelance writer, has authored numerous articles on IBM*'s data processing technology.*

Service for software

The Field Engineering Division has been giving top priority to software service. As computers become more sophisticated, so does programming. Special representatives deal with the complexities.

Today's growing customer dependence on advanced data processing systems —systems that involve complex operating systems and program products— has given software service top priority in the Field Engineering Division.

The key to software service is the programming systems representative, who does for programming what the customer engineer does for hardware.

By the end of 1974, the number of programming systems representatives (PSRs, as FE calls them) will have increased 70 percent since 1972. This will result in approximately one PSR for every five customer engineers who deal with hardware.

Richard T. Liebhaber, Field Engineering Division vice president, field support, expressed IBM's concern with software service recently at a customer seminar in Poughkeepsie. "We are urging our programming people in the field," he told the customers, "to be involved as much as possible with your programming employees." One reason for this is the highly interdependent relationship which exists between the customer's application code and the IBM-provided operating system. With the growth in operating systems, as new capability has been developed, this relationship has become complex. A System/360 operating system, for example, contained 300,000 lines of code. System control programs of the 370 sometimes contain three million lines or more.

Although there has been a significant improvement in programming code quality in general, there has been an even greater increase in the amount of code. So FE has been emphasizing new training techniques to help the PSRs find better ways to identify areas of programs where errors exist and then to manage what company resources are available—programming development people as well as FE's own field technical support centers—to fix the problems.

The field technical support centers have access to a comprehensive databank with known fixes to previously defined problems. These can be furnished speedily.

As part of the software service thrust by FE, says Liebhaber, the division's managers in the field place as much emphasis on programming service as they do on hardware service.

"The PSR doesn't have to be an expert in 100,000 lines of code," says Liebhaber, "but he must know how to go about identifying where a problem lies and getting the right assistance to solve it rapidly. That, in fact, is the challenge of program support today and it's likely to be the same tomorrow." ■

would like to see an IBM supermarket system in every supermarket in the world. We want the supermarket manager to gain access to data via his terminal so he can do a better job of managing his store.

"Three years ago such expectations would have been considered pie-in-the-sky. But today we can think about it realistically.

"The growth potential for terminals is very great," he adds. "There are several hundred thousand terminals in use today. There may well be several million by 1980."

As Akers views it, the steady migration of the computer into the operational aspects of business has been mainly responsible for bringing the casual or nontechnical user into direct contact with the machine. "At first the computer was mainly an administrative tool," he says.

"But now information processing technology has become a part of the fiber of business, itself. The computer reports what's happening on the factory floor, in the retail store, in the supermarket, in the bank. By definition, that means the computer is going to be used by all kinds of people in all walks of life."

It's wrong, of course, to suppose this trend has emerged suddenly or that it's anywhere near close to running its course. "The computer has been moving gradually into the operational aspects of business for more than a decade," says Bob O. Evans, IBM vice president and president of the System Development Division, "starting with applications in the process industries. But lately it has been advancing in that direction at a quickening pace."

The requirements of a real-time environment, Evans observes, have literally placed impossible programming demands upon the customer. "For every ten steps we take forward in hardware, programming complexity increases by a factor of a hundred," he says. "The manufacturer, therefore, must provide the programming so that the customer can move into data base/

data communications as soon as possible."

The viewpoint of IBM executives is borne out by trends in the marketplace, especially those expressed by Allen J. Burris, vice president for systems and planning at The Northern Trust Company in Chicago. Burris, like McKay and Akers, believes that computer manufacturers must focus on software developments that make it easier for inexperienced users to communicate with the machine. "Software is the sensitive spot," he says. "Right now it's the weakest link in the chain."

Burris is in a good position to interpret the needs of the marketplace. As president of GUIDE, a large IBM user-group, he comes in constant contact with IBM customers in different industries. And as an executive of one of the nation's largest commercial banks, he is in charge of implementing systems, with the future clearly in mind. Understandably, he prefers to draw from his own banking experience to illustrate his points.

Like many other organizations that generate vast amounts of data, Northern Trust has turned to IBM's Information Management System (IMS), a software program that acts like a traffic cop inside the computer, managing data bases. The bank currently runs IMS on a System/360/Multiprocessor 65 (next year it will convert to a System/370 MP 168), linked to two hundred 3270 display terminals. In short, the bank has built huge, interlinked data bases to which authorized personnel can gain access via remote terminals.

The on-line system, Burris observes, allows untrained users—from tellers to investment officers—to enter and retrieve information. "Now," he says, "they want to manipulate that information to find out what they need to know. They want to communicate directly with the computer without a technical programmer coming between themselves and the machine. But there's one obstacle—they don't know how to instruct the machine in a language it understands."

An important part of the solution, Burris feels, lies in designing systems and software so that the computer itself can translate the user's desires—once he defines them clearly—into application programs. "What we must do," he says, "is adapt the machine to man, rather than man to the machine."

The idea of "adapting the machine to the user" has been a fundamental concern of IBM for quite some time. "There are two ways we can approach that objective," says Elaine R. Bond, director of systems and programming at Corporate Headquarters.

"First, we can put more function in the machine—that is, we can write vast numbers of prepackaged programs to do what the user wants. Then the user can press the right buttons and invoke the programs at will.

"Second, we can put programs in the machine that perform translation functions. Then users can tell the machine what they want in their own language. The program will translate the users' instructions into machine language and prepare application programs itself."

How can we make it easier for nontechnical users to tell a computer exactly what they want it to do? Finding the answer to that question has been a main preoccupation of groups of behavioral scientists and computer scientists at the Thomas J. Watson Research Center. Studies conducted in the behavioral science group by Dr. John Gould and Dr. Lance Miller indicate that many of the human "hang-ups" that impede interaction between individual and machine are rooted in the differences between a "natural" language like English and a programming language like FORTRAN or COBOL.

"Natural languages," Dr. Gould observes, "depend upon context for meaning. They are often highly ambiguous, and conditions aren't always specified. A programming language, however, depends upon exact syntax. Everything must be logical, sequential, specified."

Recently Dr. Moshe Zloof of the automatic programming group has created a new "query language," called Query by Example, which offers great promise for helping nontechnical people use a computer. Rather than writing a program, the user feeds hypothetical examples of what he or she wants into the computer via a terminal.

The examples serve as sets of instructions which the computer can translate into a program to carry out its assignment. Behavioral Scientists Drs. John Thomas and Gould have shown that high school and college students with no programming experience easily learn to ask complicated questions in the Query by Example Language.

The creation of teleprocessing environments rests, of course, upon a large body of research and development work, most of which is still going on. "A whole series of innovations have occurred during the past five years," says Evans, "that have enabled us to bring data base/data communications to numerous customers.

"First," he says, "we had to design central processing units so they could cope with a dynamic, interactive multiuser environment. Then we had to come up with operating systems that would control the CPUs.

"Next we had to develop a pervasive plan, called system network architecture, that tied all elements together, and then we had to develop arrays of terminals to satisfy a variety of different customer needs.

"Finally, and perhaps more importantly, we had to develop the programs that allow customers to move swiftly into data base/data communications applications.

"For years I've called these software programs 'the glue that holds whole systems together,' " Evans says. "But they're more than that. In the case of data base/data communications, it's the programming that allows the customer to make the transition to a complex teleprocessing environment. And that transition is opening a vast new world for the user and for the people of IBM." ■

EXERCISE—W

by Lee Edson

Exercise aficionados in the U. S.—sometimes called the exercists—like to see their presidents happily engaged in physical fitness routines—like on the golf course, Jack Kennedy playing touch football, Lyndon Johnson in the saddle, and now Gerry Ford in the pool for his daily swim.

Critics of exercise—those of us who are presumably lazy and fat—look with equal pleasure to Britain's Winston Churchill, who during his lifetime almost never exercised. He drank lots of brandy, smoked big black cigars, ate fattening foods, and lived to the ripe old age of 90.

Few doctors will urge you to take up the Churchillian life-style, but the question posed by the striking contrast between presidential activity and prime ministerial inactivity is nonetheless an important one: How much do we *really* understand about the effect of exercise—particularly regular exercise—on the human body? Does it improve health? Prevent heart attacks? Increase longevity?

For years, answers to these questions have come largely from the locker-room memoirs of athletic coaches, subjective observations of medical men, and the promotion efforts of exercise bugs who have turned their own "skinny bodies into Atlas specimens." Even if the contention that exercise is good for you seemed hardly challengeable, the actual scientific validity behind it was virtually nil. With the rise in recent years of the field of exercise physiology and the development of new instruments of measurement, exercise has finally been subjected to scientific scrutiny.

HO NEEDS IT?

One of the leading authorities in the field of exercise physiology is 73-year-old Thomas K. Cureton, head of the University of Illinois Physical Fitness Institute, and a pioneer in this field of research. He became famous for predicting that Roger Bannister would run the four-minute mile on the basis of his heart size, which was 38 percent above normal. Back in the late Forties, Cureton set out to show that all people can be helped by a regime of regular exercise: young, old, and middle-aged, those fit and those in apparently flabby condition, sedentary executive types, laborers and athletes.

Professor Cureton summed up his views of the health of the body in terms of "cardiovascular fitness"—essentially the efficiency of oxygen pumping—and proceeded to determine how exercise affects it. Cardiovascular fitness is no single dimension. It is the sum of a number of components, mainly energy of the heart stroke, oxygen intake, and pulse pressure during and after work. Scientists have developed many devices to measure these components in the laboratory, ranging from the sensitive suspension table known as the ballistocardiograph, which measures the force of a solitary heart beat, to an ergometer, a stationary bicycle which can simulate a bicycle going up hill to measure the oxygen used up for work.

One important discovery is that physical training can improve the output of the heart when it is quiet or resting, thus sending more blood into circulation with each stroke than would be the case if the individual had not exercised. What

happens is that, during physical exertion, the pressure of the blood entering the right side of the heart—the venous pressure—becomes elevated. This forces blood into the heart during diastole, or expansion of the heart, stretching the cardiac fibers and boosting contraction during the systole, or contraction period of the heart. The volume of blood ejected with each beat —that is, the stroke volume—rises, sometimes as much as one and a half times over what it was in the resting state. Athletes, especially swimmers, have a large stroke volume at the height of training, and it continues to be high in the resting state as well. It is this increase of the heart's output—which may go from five or six quarts of blood per minute to between 19 and 37 quarts—that improves the body condition.

According to Cureton exercise also affects oxygen intake and utilization, both of which are related to the body's endurance for work. The more oxygen going in at a faster clip, and the more efficiently it is used by the muscles, the higher the energy output. Endurance depends on the total oxygen supply "drunk up" by the tissues, including the oxygen reserves which the body seems to call upon for last-minute exertion. Heart disease patients, incidentally, have a low oxygen intake; athletes have the highest.

The physiological effects of exercise would mean little, except to those seeking higher endurance or strength in sports, if one could not tie the results to desirable across-the-board health items, such as freedom from disease, extra vigor, and longevity. Here the evidence for exercise is largely subjective.

"We've run the program for 28 years," says Cureton of his physical fitness laboratory, "and we've never found anybody hurt by exercise. Nobody's heart was ever damaged. Exercise did a lot of good to many people, warding off disease. Man is built for heavy work—so why shouldn't exercise help him?" Others have noted that exercise seems to fend off various diseases, including emphysema and, of course, cardiovascular disease. While there is no actual proof of the part exercise plays in recovery, or longevity in general, doctors seem to feel that it is basically helpful.

Physiologist Sid Robinson of Indiana University, former co-director of the World War II Harvard Fatigue Laboratory, and his colleagues followed a select group of 16 champion runners, including miler Glenn Cunningham. They were contacted 20 to 30 years after their moment of glory and reexamined with regard to lung volume, body density, and percentage of fat, as well as the usual physiological functions such as oxygen intake, heart rate, and blood pressure. Robinson found that their life-style—in particular smoking, drinking, and exercise habits—and not their training in youth, determined what happened to these athletes phys-

---

Do you puff and pant after climbing stairs? Do you feel dizzy after a couple of sets of tennis? That could be a sign of "oxygen debt"—a condition created by bursts of speed and effort that makes you gasp for air after such activity. You might be a candidate for a cardiovascular attack.

Companies in the United States are becoming increasingly concerned about the alarming number of cardiovascular attacks and other effects of stress upon their employees. Many firms have set up exercise programs.

IBM has long maintained recreational facilities at major locations, but according to Dr. John C. Duffy, the company's medical director, recreation is not designed to recondition the human body, it is a leisure-time change of pace.

Back in 1970, Dr. Duffy set up guidelines for establishing supervised voluntary exercise programs in IBM.

HOW IBMERS KEEP FIT

by Ruth S. Baker

"In a sedentary job, especially," he says, "it's almost impossible to get sufficient exercise, to keep fit, to relieve office pressure, unless you set up a program of regular physical fitness exercise on your own time and stick to it. That doesn't mean just tennis or golf on weekends. We are trying to encourage people to create their own daily regimen and self-discipline."

In 1973, the Advanced Management School and Executive Program at Sands Point, Long Island, started their formal Physical Fitness Programs. Students volunteer, with their personal physician's approval, to jog and do calesthenics at 7 in the morning, three times a week.

"A lot of people don't like to exercise alone," says Dave Rodes, professional director of the fitness program, "so we use the class approach. It gives a sense of 'we're all in this together,' but each exercises at his or her own pace, with

ically. Those who had not continued vigorous physical activity or conditioning were in the same poor physical shape of the average businessman of their age. In fact, two champion athletes who had led more sedentary lives were in very poor physical condition and had suffered heart seizures.

James (Doc) Counsilman of Indiana University, who has trained more champions (including 1972 Olympic star Mark Spitz) than any swimming coach, has followed the lives of a number of champion swimmers. On the whole, says Counsilman, those who keep up with swimming are in better all-around shape than those who fall into sedentary habits.

Doc Counsilman is his own best study. A champion marathon swimmer at Ohio State, who competed until he was 28 and weighed a trim 180, found that after he left competitive sports he was growing flabby. At 50, he weighed 250 pounds, had difficulty breathing, and suffered from backache. "I went back to swimming," he recalls. "In two years, not only have I lost 60 pounds (diet went along with this), but I feel better than I ever did." Counsilman now swims two miles a day and is hardly winded when he leaves the pool. His aches have gone, he says, and his systolic blood pressure of 135, has dropped to a youthful 120.

Not everyone agrees that a regimen of exercise of the kind that builds athletes is good for every human body. Dr. F. E. H. Lightbody, for example, a physiatrist (a specialist in physical medicine) at Greenwich Hospital in Connecticut, who uses exercise in rehabilitation work, says: "People differ so much that it's difficult to make blanket statements about exercise that apply to everyone."

Dr. Peter Steincrohn, a critic of what he calls the "exercise mania" in the U.S., maintains that sedentary people exercise more than they realize, walking, and climbing stairs, for example. He emphasizes that reduction of tension and appropriate relaxation are more important than exercise per se. But most doctors agree that moderate exercise does, in fact, reduce tension. So the exercists still come out ahead.

Even Dr. Steincrohn recommends swimming and jogging, "which is the gentlest gait and produces several beneficial effects if we use it in moderation." Jogging seems to force the body to become conditioned to an increased need for oxygen. When the body meets this need through exercise, the heart and breathing system become much more efficient. Some proof—Dr. Herbert deVries of the University of Southern California ran a program of jogging for people from 50 to 87, at a moderate pace, to be sure. His major finding: Even people in their 80's improved significantly in vigor. ■

Lee Edson, a frequent contributor to Think, *is the author of three books. He writes for* The New York Times Magazine *and other major publications on scientific and medical subjects.*

---

no competition. At the same time we create the discipline to carry on alone later.

"Also, some don't care for jogging —the 'in' exercise today—so swimming, cycling, sustained climbing, or hard-stride walking can be substituted. We tell them not to be slavish to any one thing."

The program is concerned mainly with the "aerobic" effect in exercise. That means oxygen reaching the muscle cells while the major effort is taking place, thus better sustaining the activity. This type of exercise is described in Dr. Kenneth H. Cooper's book, *The New Aerobics,* officially adopted by the U.S. Air Force and a basis for the Sands Point programs. Emphasis is put on big leg muscle exercises as they provide more sustained effort. Full circulation, deep and steady breathing, is what must be achieved for cardiovascular fitness.

Cardiovascular fitness is one goal, flexibility and suppleness, strength and tone of muscle are others. "We give pointers on how to care for backs to avoid injury," adds Rodes, "and how to stretch muscles that tend to shorten from the inactivity of office life."

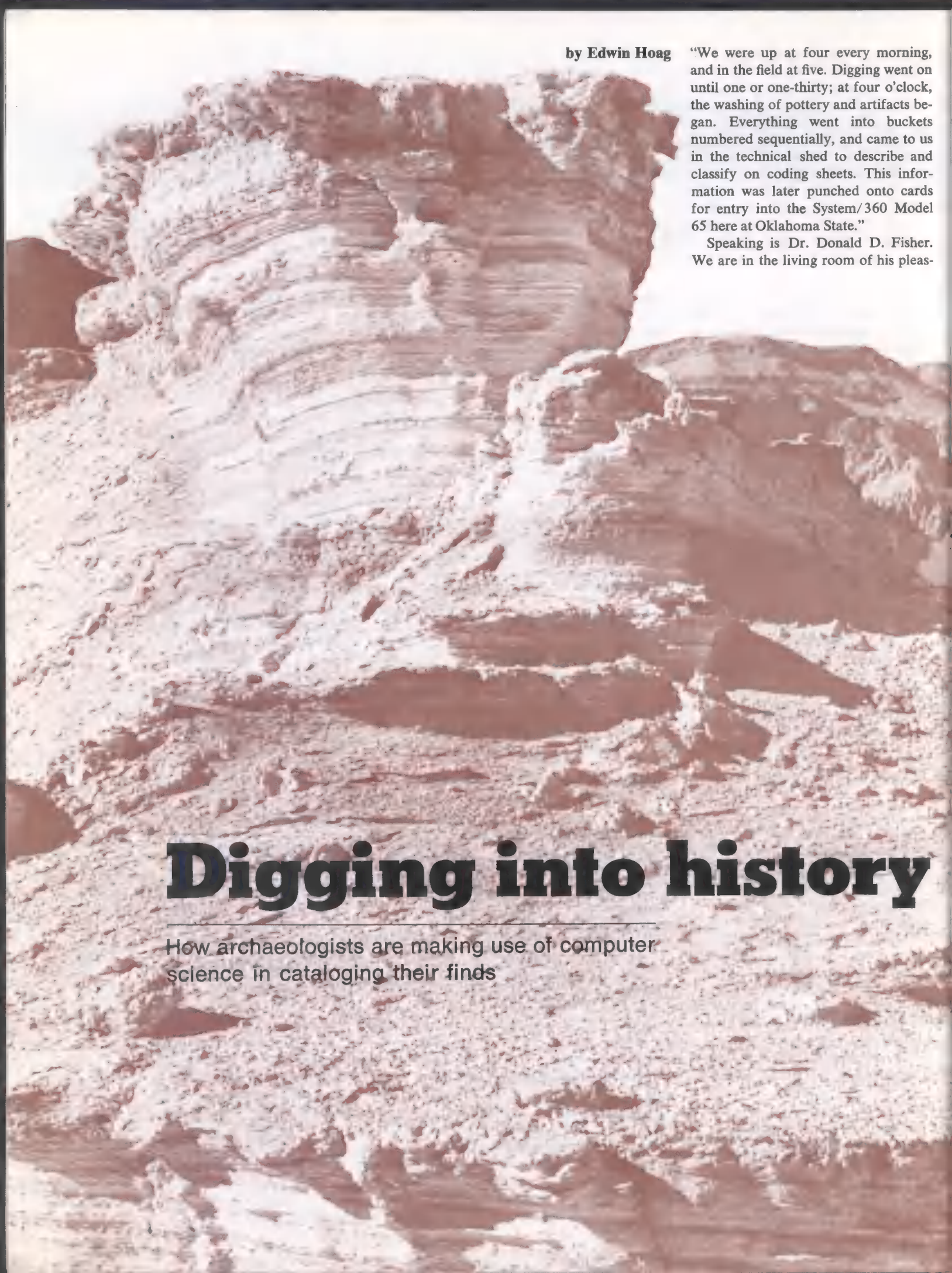
Along with this, eating habits are important. "For instance," he says, "if you have a strawberry shortcake with 400 calories, you'd have to spend 308 minutes doing nothing; 77 minutes, walking; 49, bicycling; 36, swimming; or would have to run 21 minutes straight to 'work it off.' "

At a similar voluntary physical fitness program in the Middle Management School (also at Sands Point, and at Glen Cove), Donald I. Gorr, the school's program manager, says: "Our program just started October 7th. We have the same professional direction, but, of course, in just one week (as compared with three at AMS) we can't get people in shape. We just want to show them that physical fitness is important, through lectures and literature, and hope to give them the information and incentive to carry on."

In Southbury, Conn., the one-week New Managers School has an informal sports and exercise program—and take-home literature. And some IBM Clubs around the country sponsor informal exercise activities for employees and their families.

"However," sums up Dr. Duffy, "no company-sponsored program can do it for you. Whether you jog, swim, do ballet warmups, or whatever, the important thing is to keep doing it—regularly. Make it a part of your everyday life.

"Company programs for employees can teach the *abc*'s of physical fitness; but you must continue on your own if they are to have any lasting value. You have to volunteer to keep healthy." ■



Digging into history

How archaeologists are making use of computer science in cataloging their finds

by Edwin Hoag

"We were up at four every morning, and in the field at five. Digging went on until one or one-thirty; at four o'clock, the washing of pottery and artifacts began. Everything went into buckets numbered sequentially, and came to us in the technical shed to describe and classify on coding sheets. This information was later punched onto cards for entry into the System/360 Model 65 here at Oklahoma State."

Speaking is Dr. Donald D. Fisher. We are in the living room of his pleas-

ant home a few miles beyond the perimeter of the Oklahoma State University campus in Stillwater. A cordial man, who is director of the Oklahoma State School of Mathematical Sciences and professor of computer sciences, Dr. Fisher has spent over 20 years in computer sciences. He is telling about his participation in the continuing archaeological expeditions to the buried ancient city of Caesarea Maritima on the coast of Israel between Haifa and Tel Aviv. He went on the expedition to develop a computer-based information system for recording and analyzing all data relative to thousands of artifacts being unearthed at the site.

As a slide projector clicks and the image of a fortress-like structure flashes on the wall, the slim, 44-year-old professor comments: "That's a Crusader castle from around the Tenth Century. It was excavated several years ago and restored. It's right on the coast, overlooking the Mediterranean. That's a harbor out there, yet to be excavated. It held the Roman fleet in the heyday of the empire."

The projector clicks again, a new image appears on the wall. "That's part of the self-flushing sewer system built during the Roman period. You could drive a truck through it. And just behind that is an amphitheater not yet excavated. Initial probes indicated it was large enough to hold 30,000 people. King Herod, who ran Palestine for the Roman Empire, built the city between 22 B.C. and 10 B.C. It became the Roman capital of Palestine and later survived through Byzantine, Arab, and Crusader occupations. In this dig we went down to, but barely penetrated, the Roman period. That took us down about twelve feet." The projector clicks again, and again, . . .

"That was C Site," Dr. Fisher says, gesturing toward the slide image of archaeologists and students working in

Dr. Donald Fisher, director of Oklahoma State University School of Mathematical Sciences and professor of computer sciences, explains layers of "digs" at the ancient city of Caesarea Maritima on the coast of Israel.

one of the square pits of the dig. "There were mosaics on the floor and walls. It was a Byzantine building of some sort. Notice the students with dental picks and brushes going through skeletons. They found about 80 of them in that layer. They have to be handled very carefully, as does everything. It was definitely a destruction layer, indicating the city had been sacked or burned.

"In another square, Site A, we pulled out 30 or 40 ceramic lamps, just a few inches large, and ranging in time over the Roman, Byzantine, and old Arabic periods. We found plates, hair pins, glassware, pavings, coins, jars, beads, knife blades, bowls, a frying pan handle, even ancient dice, and a great deal of pottery vessels and sherds, that is, broken pieces of pottery. One of the big finds was five miles of collecting tunnels in the mountains, the water source for the aqueduct that served the city. It's hewn in solid rock."

The last slide shown, Dr. Fisher reaches for a wall switch, lighting the room. "It was a tremendous learning experience for me—I knew very little about archaeology before. And it was an extraordinary experience for the students who went along. There were eight colleges, universities, and other institutions represented in our group; this summer there were approximately 20.

"Robert Bull, formerly director of the American School for Oriental Research, now at Drew University, got the dig permit from the government of Israel. He asked James Kirby and Kyle Yates, Jr., of the Oklahoma State faculty here, both of whom have worked in archaeology, to join him. The question of handling the data arose, and that's where I came in. Essentially, it's a problem of describing and classifying artifacts, relating them to specific locations, then creating a data base—one that can be used in a great variety of ways for analysis and study. It's not terribly sophisticated from the standpoint of computer science—sorting, merging, collating, analyzing—but from the standpoint of the archeologists it's extremely valuable. Bull has said that we will have, for example, the best-documented collection of Palestinian pottery in existence."

Dr. Fisher asks us to meet him the next morning in his office. It is in the O.S.U. Mathematical Sciences Building. With its desk, tables, book-lined wall, and blackboard, it is a fairly typical academician's office, but with this distinction: Boxes of punched cards and stacks of computer printout fill one shelf and, next to them, several boxes of pottery fragments, carefully marked for identification. Dr. Fisher steps to the blackboard and quickly sketches the outline of a typical square in an archaeological dig field. "Each square is ten meters by ten meters, but you leave a two-meter ridge of land (balk) between them to walk on, so they are actually eight by eight. Everything has to be recorded as you dig down. Everything is measured, and photographed, and sketched by an artist. Then the balk is tagged and pinned so you know later what came from where. All digs are done stratagraphically. It's important to archaeologists to know what finds are associated with particular strata. It gives them clues to where to dig next.

"For the first ten days, before buckets were filled, I helped with the digging. It's hard work down in the pits, hot and still. Then, for four and a half weeks, we worked in the tech shed with the bucket contents, completing coding sheets—the wind blowing, the sand getting in our eyes."

Dr. Fisher picks up an orange publication from his desk. "This is the coding manual we produced here at the University before we went over; it tries to simplify the procedure for non-computer people. It's based on an earlier one developed by two scholars at Waterloo Lutheran University, one which we modified substantially here and then later revised over there at Caesarea. We coded artifacts and sherds by identification number, location in terms of field, square, bucket number, elevation and stratum, material, function, technique of manufacture, Munsell color number, time period, photo number, and the scale of drawings that were made. For sherds we also recorded preserved features, completeness, stance, measurements, shapes, grit size and frequency, pro-

duction method, incisions, decorations, and colors of paints or glazes. We couldn't keep up with the buckets in our coding, even though several of us were doing it. But I came back with a couple of reams of coding sheets. They produced, in turn, some 10,700 punched cards."

Gesturing toward the large stack of printout on a nearby shelf, Dr. Fisher continues: "We've generated several computer catalogs with details on 2,500 sherds and 1,250 artifacts selected there for classification. One catalog has detailed information for each item. In another catalog is a summary of artifacts and sherds by locus number. The point is, we can get into the data base and ask any kind of question. We can pull out information by location, or characteristic, or historic period. We can search the data for specific artifact or sherd associations. The coding system is thorough: Using the descriptions and categories in the manual to describe just the rim of a vessel, for example, we have calculated you can describe 186,000 distinct rim shapes.

"Another thing we have in mind is the use of plotters. We have all this descriptive information from the coding sheets in the system. Using a plotter, we can put into the system the characteristics of these vessels—the rims, curves, and the like—and it will plot them out, show us what they looked like. We hope to have graduate students work on that."

Returning behind his desk, the professor nods at the boxes of pottery fragments nearby. "We're now in the final coding stage of those sherds.

"I hope that some students in our computer and archaeology course can go to the dig one summer. They will be coding sherds on-site; it will make the course more meaningful. They'll be learning about computers and archaeology in real-time, you might say." ■

Edwin Hoag, a freelance writer, specializes in articles on technology, history, and the humanities.

Speak Up

The program celebrates its 15th anniversary this year. It gives employees the chance to question any IBM executive about any aspect of the business.

What celebrates its 15th anniversary this year, has produced more than 140,000 letters, and thrives on anonymity?

It's IBM's Speak Up program.

Started on an experimental basis at the San Jose plant in December 1959, the program has spread throughout the United States and World Trade countries.

And if imitation is the sincerest form of flattery, then IBM's Speak Up program has been flattered. Several hundred other companies have sought information about this straightforward communications channel, and many have put their own versions of it into effect.

What do outsiders find most intriguing about the program? Perhaps the element that arouses the most interest is the fact that any IBM executive can be questioned about any aspect of the business, and will take the time to provide a straight answer, where possible within ten working days.

The IBM program is administered in the U.S. and other countries by 87 Speak Up coordinators. It is their job to see that each signed letter from an employee is answered by an authority on the subject, and that the identity of each letter writer is carefully guarded —known only by the coordinator, unless a personal interview is specifically requested by the writer.

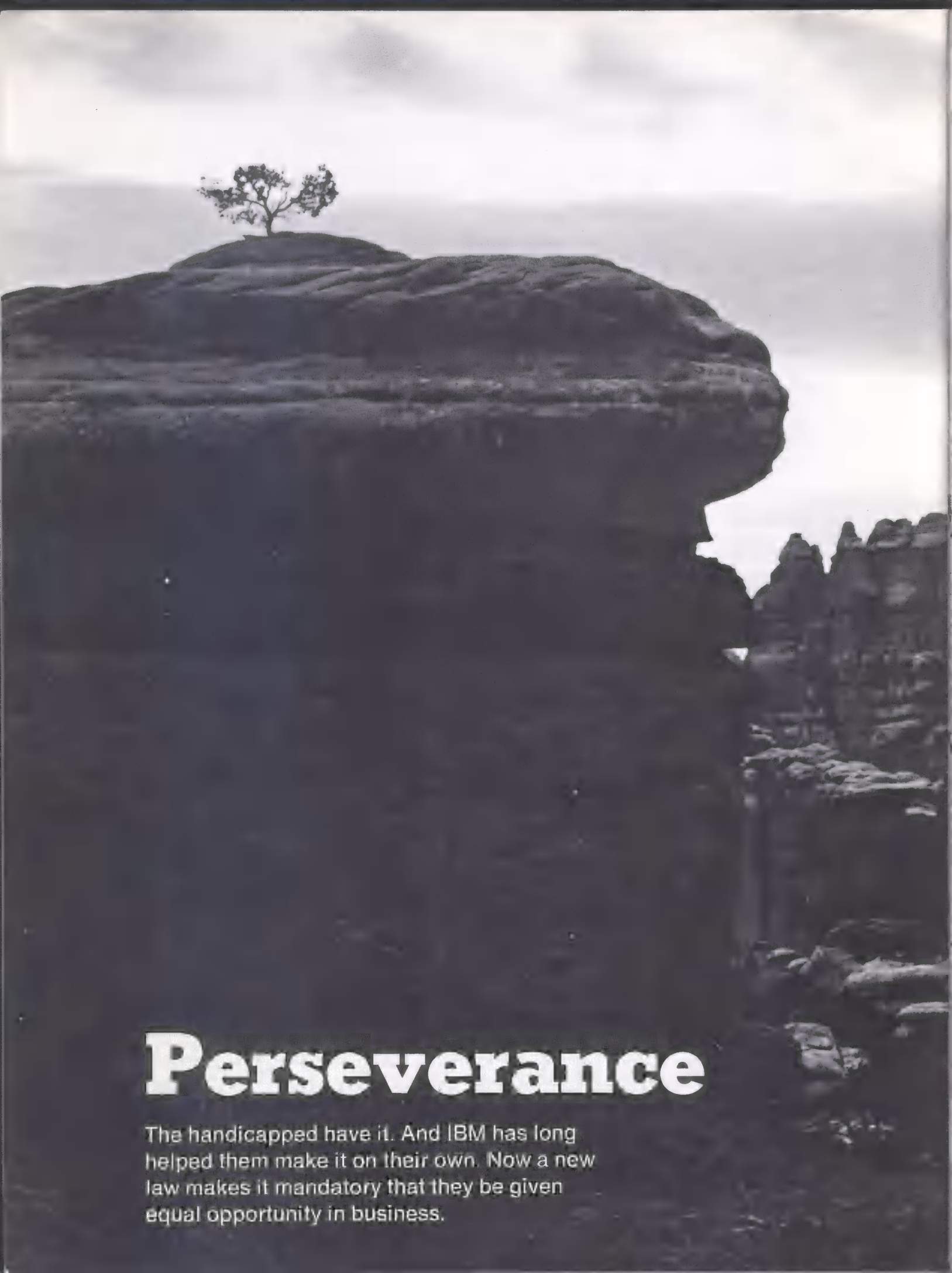
An indication that IBM employees believe that they can ask questions or express their viewpoints (often critical) without their identities being divulged, is the fact that about 90 percent of the letters are signed.

While the name stubs that go with Speak Up letters formerly were destroyed once an answer had been sent out, they are now temporarily retained in sealed cartons at IBM's record center in Rotterdam, N.Y. This is because of a judicial order barring IBM from destruction of documents. However, it has no effect on confidentiality and the stubs will be destroyed when the court order is lifted.

Over the years, the Speak Up program has given IBM employees a chance to sound off on issues, large and small. It has alerted management to policies and situations that need changing. And, it has given management better insight into the morale of IBMers everywhere.

Still, in monitoring the processing of Speak Up letters, Corporate communications has found that the program is far from perfect. Some questions take too long to answer; responses are sometimes defensive or less candid than they could be; action is promised but not always taken. In a poll (anonymous) asking for reaction to Speak Up answers, one writer said: "My reply was a letter full of ambiguous platitudes which told me nothing substantial or valuable."

Among the favorable comments, which the company believes is more indicative of how most IBMers feel about the program, was: "Instant action was taken and the 'wrong' was righted. Thank you very much. Speak Up works." ■



Perseverance

The handicapped have it. And IBM has long helped them make it on their own. Now a new law makes it mandatory that they be given equal opportunity in business.

Programmer-analyst Phil Bravin looks much like any other IBMer in his office in Englewood Cliffs, N.J. The trimly tailored, athletic-looking young man begins his day, briefcase at his side, driving to see his customers.

But Bravin is different. Call to him down the hall and you'll discover why. He doesn't turn around to respond. He hasn't heard you. Phil Bravin is deaf.

Though once classified as a "deaf-mute," Bravin is anything but mute, as his fellow workers and customers are quick to say. They find him a lively conversationalist, a warm and witty man, with a ready humor always close to the surface. His speech is soft, difficult for a stranger to understand at first (he keeps a paper and pencil at hand, as he says with a smile, "in case of fire") but quickly intelligible.

Bravin is an example of how personal determination and effort, with the support of management, can lead a capable physically handicapped person along a satisfying career path.

IBM has a long history of encouraging employees such as Bravin. But now recruitment and promotion of the handicapped have been given increased attention with the passage of the Rehabilitation Act of 1973. This Federal law has an impact similar to that of the Federal Contract Compliance Program created by a Presidential Executive Order in the early 1960s. This program required affirmative action on the part of Federal contractors in the employment of minorities and women. The new law makes it mandatory for all firms who have contracts of $2,500 or more with the Federal Government to hire and advance the qualified handicapped.

At present, fewer than half of the 12 million employable handicapped are working, according to the latest census data. Many lack job skills and require training. The Rehabilitation Act will help fund their training through state and private rehabilitation agencies. It will then be industry's turn to give them a fair chance at jobs.

To facilitate hiring procedures, an Affirmative Action Plan has been outlined by the U.S. Department of Labor. Among its key points: Companies doing business with the Federal Government must state in their advertising that they employ the handicapped; they must make reasonable accommodations for the employment of the handicapped which will reduce the effect of the handicap and enable the employee to do his or her job; companies must file an Affirmative Action Plan and submit an annual progress report each March 31.

In reviewing the qualifications of a handicapped person, IBM managers are being encouraged to look beyond the *dis*ability to the *a*bility. By judging the qualifications as a whole, the manager can best match the person to the job. For example, an IBM manager recently hired a blind secretary because she has the skills needed: typing, answering the telephone, and the ability to use IBM equipment.

IBM is studying the problem of design changes in an attempt to see what accommodations might be made for the handicapped.

In this respect, technology has been a boon, with special equipment to lessen the impact of disabilities and to speed and ease work operations: electrically operated doors, push-button phones, dictating equipment that feeds instructions on the asembly line to assist the blind, a paddle wheel to replace the platen turner that moves the linespacer on IBM Selectric Typewriters, a template that serves as a finger guide to accurate typing, and a wrist restboard developed by the Office Products Division that fastens on the frame of a typewriter so the fingers can function with arm paralysis.

Once hired, a handicapped person often faces another possible barrier: job mobility. Many become proficient, and both manager and employee might be hesitant to consider job changes.

In certain areas of computer technology, specific handicaps have become conversely, assets. For instance, in programming both blind and deaf, after overcoming initial obstacles, have often proved themselves to be competitive and capable of great concentration.

It is not unusual for a handicap to lead to increased production for an entire department.

One example occurred on an assembly line. The manager recognized that a blind worker could be helped to a better performance. Another blind worker designed a jig for him. It worked so well that the new worker's production shot way ahead of others on the line. Sighted workers were so impressed they requested the same device. This resulted in increased productivity for the entire line.

Although regulations for the Affirmative Action Plan were effective June 11, 1974, detailed guidelines are still in the formative stage.

Although it is not precisely clear who would be considered handicapped under the new law, according to a modified U.S. Labor Department definition, today approximately 5 percent of IBM's total work force is classified as handicapped.

An illustration of IBM's commitment is the Boca Raton, Fla., plant, which has received the Governor's Employer-of-the-Year-Award "in recognition of outstanding achievement of the handicapped."

Walton E. Burdick, IBM vice president, personnel plans and programs, spoke before the President's Committee for the Employment of the Handicapped at its annual meeting this year. "Certainly handicapped individuals are unique—as every human being is unique," Burdick said. "And business should judge that uniqueness as a human condition and deal with it on an individual basis."

Responsibility for implementing the guidelines in IBM has been given to the Equal Opportunity department. Walter A. Smith III, formerly manager of EO program control, is manager, handicap programs, and John Seeley is program administrator.

"Over 30 years of experience," says Smith, "has convinced IBM management that qualified handicapped individuals, regardless of the severity of impairment, are a long-range asset to the company.

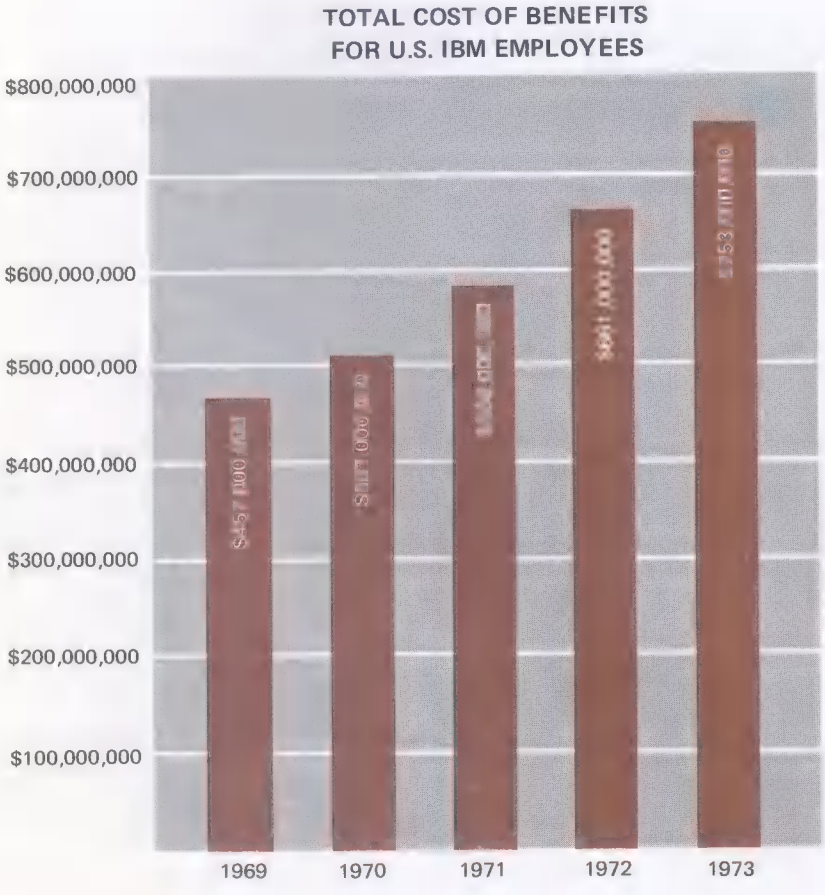
"When selectively placed, they prove to be as eager to reach job objectives, develop necessary skills, and show a team spirit, just as any other well motivated employee. Overall, the company has made substantial progress in hiring the handicapped, but we've still got a long way to go."

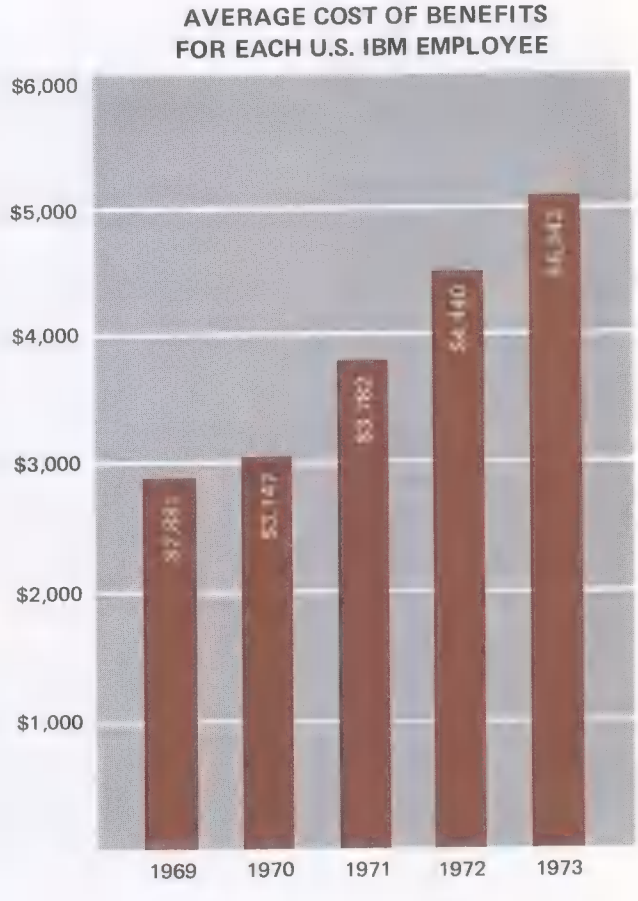
Seeley, himself a double arm amputee with a varied industrial background, adds: "Affirmative Action means a real breakthrough in job opportunities for the handicapped. Too often 'token' handicapped were hired by industry. It has been an emotional tug-of-war. The need for government action was long overdue.

"Now there's official legal recognition that hiring the handicapped is not only fair and right, but the expected route for industry to follow. The plan should be viewed not just as an obligation, but as a way for companies to train more good people." ■

$5,043

That's what it costs IBM for benefits, on the average, per U.S. employee





IBMers may not think about it very often, but company benefits provide a significant hedge against inflation. Last year, IBM's benefits bill for U.S. employees totaled a not inconsiderable $753-million. That's more than the company spent for research and development ($730-million); more than IBM's cash dividends to stockholders ($654-million); in fact, only 215 companies out of the 500 on *Fortune's* list of the largest U.S. industrial corporations had total revenues larger than IBM's benefits bill.

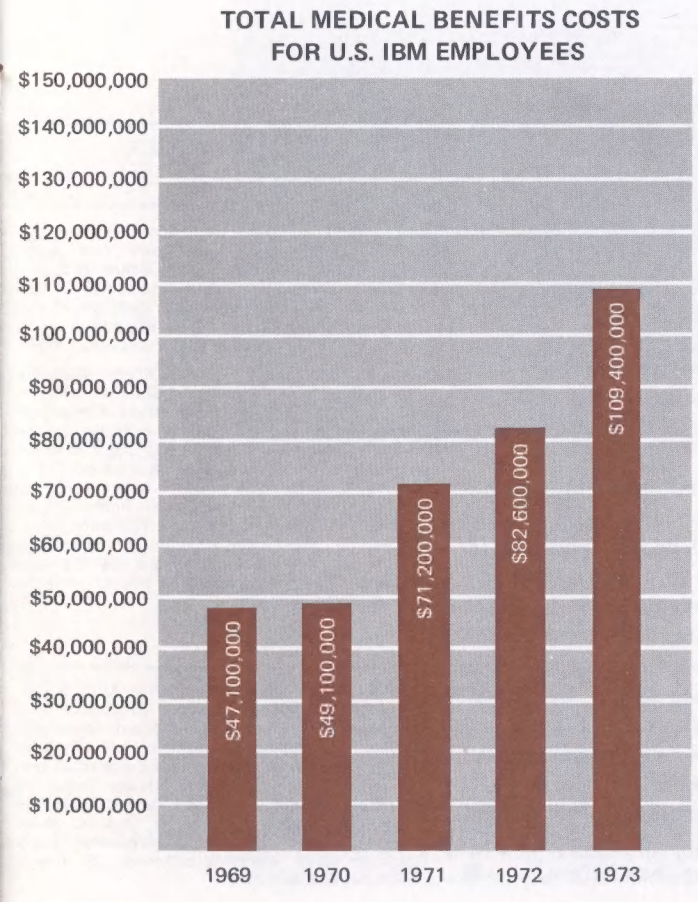
And IBM benefits are noncontributory, meaning that no part of the payment for them comes out of employee paychecks.

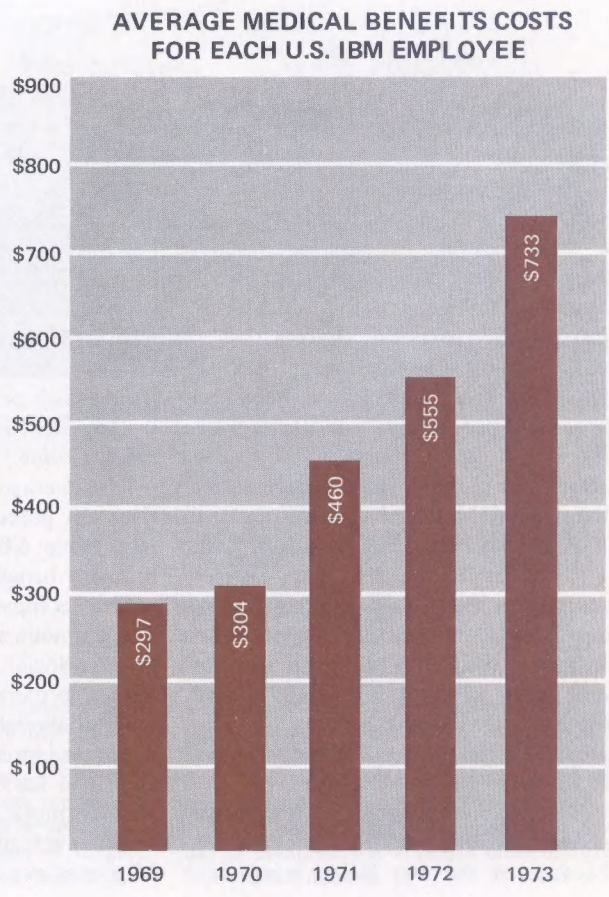
The 1973 benefits bill averaged $5,043 per U.S. employee compared with $2,881 in 1969. The bill has gone up for three reasons: rising costs—typical charges associated with a day's stay in a semiprivate hospital room that were $88 as recently as 1970 are $118 today; rising salaries which continually cause the costs of the several benefits related to salaries, such as survivor income, retirement, and time-off benefits, to rise; and, a constant broadening of coverage which reduces after-tax, out-of-pocket expenses for IBMers.

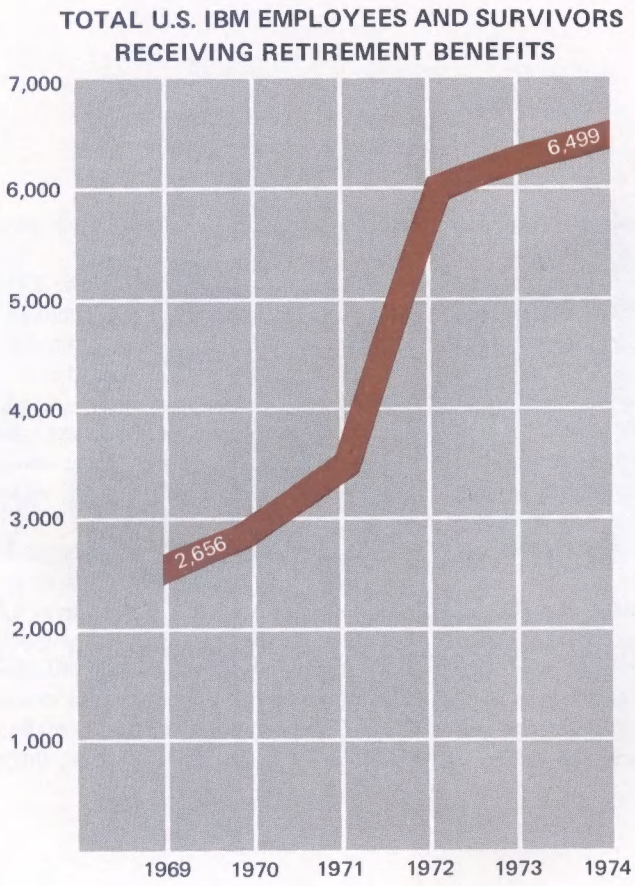
The Survivor Income Benefit, for example, was increased to three times annual earnings to an employee's survivor from two times earnings in 1972. This reduces the need for more personal life insurance and the outlay of more after-tax income in premium payments.

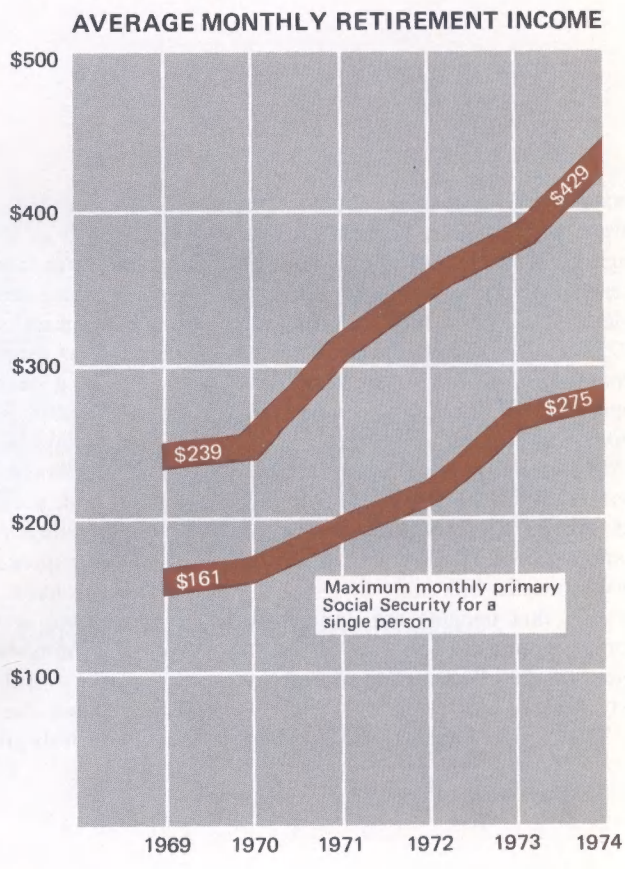
Another example: In 1970, the Sickness and Accident Plan was changed from six months to a year, providing full pay for an employee forced out of work through illness or injury. Again, the increased protection reduces the worry that employees may have about continuing to receive income while sick.

The strength of IBM's benefits package, however, doesn't rest on a single plan—but rather on a broad array of benefits and the frequency with which they are improved. (A history of benefits improvements appears in the *About Your Company* booklet recently mailed to IBMers.) In the medical area, there









have been four major improvements since 1969:

- Full coverage for emergency treatment within 24 hours of an accident;
- Full coverage under the Family Surgical Plan, rather than the former schedule of allowances that often didn't provide 100 percent of charges;
- A reduction in the Major Medical deductible from $200 for an individual and $400 per family to $150 per family. And coverage was increased to pay 80 percent of the covered charges, after the deductible, rather than 75 percent;
- A Dental Plan, introduced last year.

Such improvements, plus higher costs, have boosted the company's total cost for medical benefits 132 percent—from just over $47-million in 1969 to more than $109-million in 1973. And the cost of medical benefits per employee has jumped 147 percent—from $297 in 1969 to $733 last year.

The population group perhaps most concerned about inflation is retirees because of the way rising prices impact their already reduced incomes. As a consequence, IBM has made significant and frequent improvements in the Retirement Plan to help pensions keep pace with the cost of living.

Since 1969, this includes:

- Two increases in benefits based on length of service;
- Two updates of the earnings base from which pensions are calculated;
- One improvement in early retirement income.

The average retirement income has risen 80 percent since 1969, and the more than 6,000 people receiving retirement benefits now average $429 a month compared to $239 in 1969. These amounts are, of course, in addition to Social Security benefits which have also increased during this period.

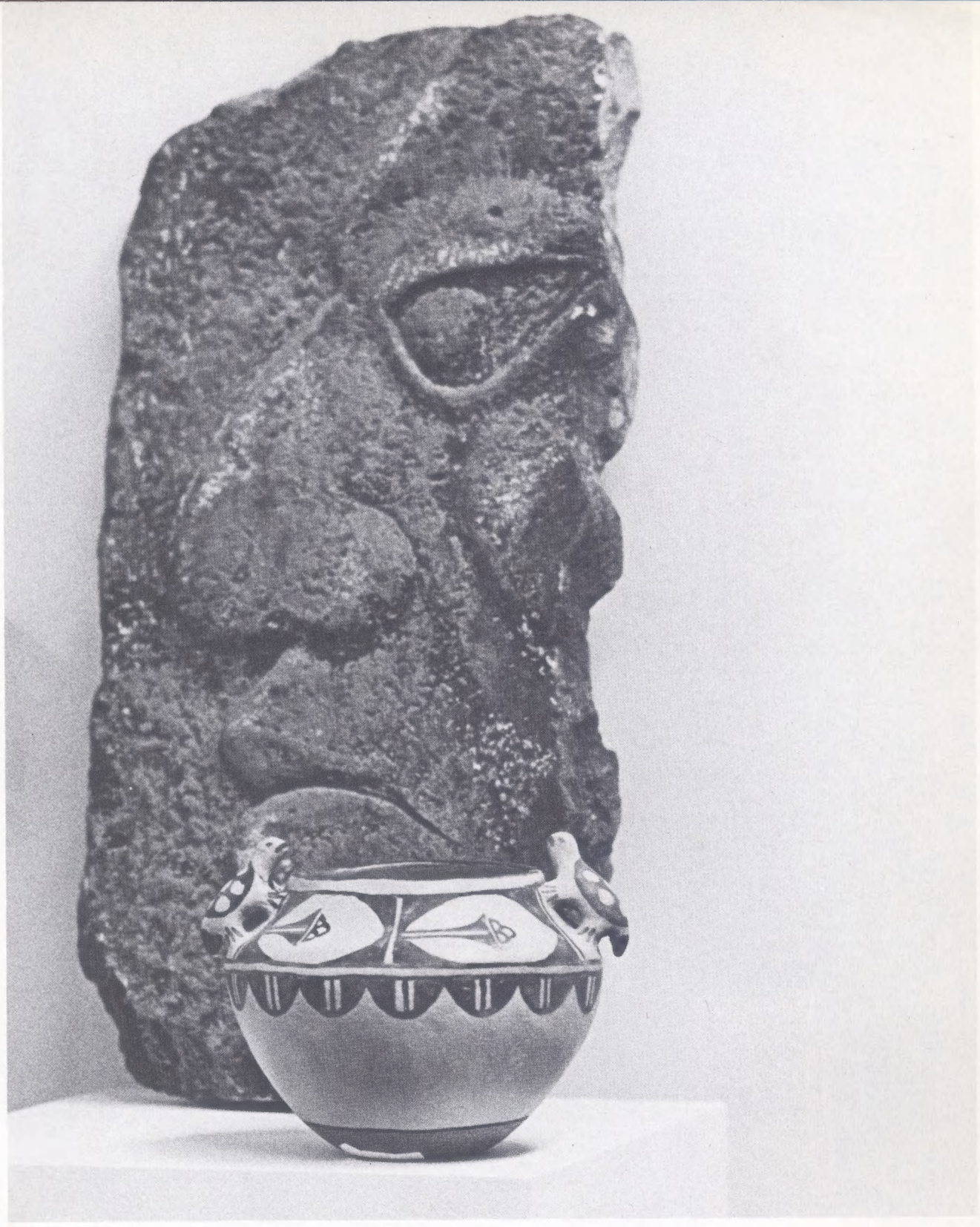
Even though the cost of most services has increased dramatically, IBM's expanded and improved benefit plans provide more coverage to employees, on the average, which means less out-of-pocket expense than before. ■

Think OCTOBER/NOVEMBER 1974

Think is published by International Business Machines Corporation, Armonk, New York 10504. Frank T. Cary, Chairman of the Board. It is produced by IBM Corporate Communications, Jane P. Cahill, IBM vice president, communications. *Think* is distributed to all IBM employees in the U.S.A., and to a number of IBM employees abroad.

Chet Hansen, Editor; Woody Klein, Managing Editor; Benito P. Carucci, Art Director; Quentin W. Johnson, Manager, Publications Production; Angelo M. Donofrio, Special Features Editor; Howard K. Janis, Senior Editor-at-Large; Geoffrey D. Austrian, Ernie Bauer, Editors-at-Large; Claire Stegmann, Senior Editor; Ruth S. Baker, Associate Editor; Walter H. Wszulkowski, Photo Editor; Elizabeth W. Brown, Circulation. The word Selectric and all other similarly capitalized marks identify products manufactured by the IBM Corporation and distinguish them from similar products manufactured by others.

Credits: Cover: Joe Barnell. Patrick Altar: 9. Barnell: 22. The Bettman Archive: 20 bottom. Tom Bocchino: 27 top right. Richard Weymouth Brooks: 44. Compix: 11. Vittorugo Contino: 12-15. Mike Ehrenberg: 30, 32. Bradford Burgher: 27 bottom. IBM Japan: 27 top left. Image Group: 26 top. Mark Joseph: 6, 10. Shelly Katz: 42. Bob Lerner: 18, 19. 25. Mastandrea Studios: 46-48. Donald L. Miller: 36-39. Rapho-Guillumette—Louis Goldman: 40-41. Walter Wszulkowski: 26 bottom, inside back cover.



Stone sculpture by Carlos Venegas, purchased from the Mexican American Art Studio, Los Angeles, Calif.

Pottery bowl by Seferina Ortiz of the Conchiti Pueblo in New Mexico.

It's not the Museum of Modern Art but Data Processing Division Headquarters in White Plains. It played host recently to a minority art exhibit by painters, sculptors, and other artists. The exhibit, which lined the first-floor corridors, is part of DPD's affirmative action program. "We're not in the art business," says Robert T. (Bob) Griffin, Jr., manager of DPD equal opportunity programs. "The exhibit is just one method of making people aware of the aspirations of minorities and their potential." The Minority Art Program started out with three paintings two years ago. "Since then," says Jack Steers, program manager of affirmative action programs and curator of the collection, "we've acquired some 50 pieces—sculptures, drawings, photographs, prints, and craftwork."

So far, all or parts of the collection have traveled to a number of communities in larger cities around the country and to eight IBM locations.

International Business Machines Corp.
Armonk, N. Y. 10504

To make sure you get Think without interruption, inform your manager, payroll, or personnel department immediately of any change in your address. If any IBM household is receiving more copies than required, send the extra mailing label to your local PDS department or, at a field location, to your administration manager. It will be removed from the circulation list.

BULK RATE
U.S. POSTAGE
PAID
IBM CORP.